(Hello World)

THE MAGAZINE FOR EDUCATORS TEACHING COMPUTING AND AI

CS EDUCATION IN A BRAVE NEW WORLD
Teaching students to question, not just execute

STOP DETECTING, START TEACHING
Tools to teach critical thinking

DON'T GIVE ME THE ANSWER!
The role of feedback literacy in strategic AI use

Issue 30 | July 2026 | helloworld.cc

CRITICAL THINKING IN THE AGE OF AI

PLUS **AI IN ASSESSMENT** · CT WITHOUT COMPUTERS · **PD AROUND THE GLOBE** · STEM OUTREACH · **WEIRD AI** WHAT WE'RE READING · **AP COMPUTER SCIENCE A** · TOUCH TYPING · **COOLEST PROJECTS** · CODE/ART **EPISTEMIC PLANE** · COGNITIVE LOAD · **SOCIAL MEDIA LESSON PLAN** · BEBRAS · **GENAI IN EDUCATION**

Raspberry Pi Foundation

Experience CS

A free integrated curriculum for computer science

Experience CS empowers educators of elementary and middle school students (aged 8 to 14) to teach computer science through a standards-aligned curriculum that integrates CS concepts into core subjects like maths, science, languages, and the arts.

Created by educators for educators, Experience CS includes:

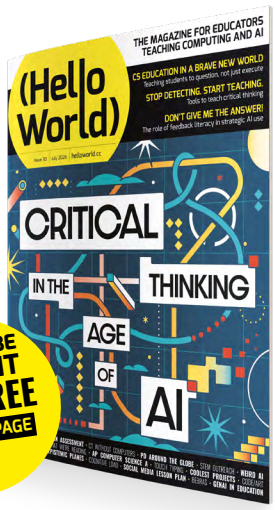
- Ready-to-use lesson plans, educator resources, and classroom materials.
- Creative projects using a version of Scratch built especially for schools.
- Simple and intuitive learning management features to track students' progress and manage classroom assignments.
- Professional development opportunities to help you feel confident teaching CS. No prior experience needed.

Discover more at: rpf.io/exp-cs-hw30

HELLO, WORLD!

The speed and ease of productivity that AI offers is enticing, so why bother with critical thinking? As someone who has always loved learning, my opinion is that critical thinking feels good. Solving a puzzle, sharing a new perspective, and doing a little digging make me feel like I've achieved something. As educators, this is one of the gifts you can give your students through teaching critical thinking skills. But in the age of AI where the answers are so irresistibly easy to access, how do we teach young people, and ourselves, to think critically?

This issue of Hello World aims to help. There's an excellent overview of the Brookings Institution report on the current landscape of benefits and risks of generative AI in children's education (pages 16–17). Please make time to have a go at two free AI detection games from Sam Illingworth (pages 26–27). I thought I'd be really good at it, but I was a thought-provoking average. And Rehana Al-Soltane explains why feedback literacy plays an important role in teaching young people how to use AI discerningly (pages 40–41). We've also revamped our news section to bring you the latest globally relevant updates in areas such as curriculum, pedagogy, and more.



SUBSCRIBE IN PRINT FOR FREE
TURN TO PAGE 83

Those with a sharp eye will notice we have a new tagline on the cover. We are now the magazine for educators teaching computing and AI, because all educators will need to integrate AI into their teaching just as AI becomes more integrated into our everyday lives. Let's get in there early with teaching critical thinking skills to young people, to not only actively engage in learning in an age of AI, but to reap the feel-good rewards too.

Meg Wang
Editor

FEATURED THIS ISSUE

BECCA ROSE

Becca introduces us to flatgames and how this corner of indie game culture built on handmade, intimate stories can unlock unexpected and personal making. Read more on pages 64–65.

PONGPRAPAN PONGSOPHON

On pages 46–47, Pongprapan explains how learners can use micro:bit, Python, and machine learning to create predictive classroom systems.

BEATRIZ PADÍN AND ELENA PONCELA

Beatriz and Elena combine physics and computer science for a 'Physics meets Technology' project for secondary students on pages 42–43.

(HW)

Hello World is the official magazine of the Raspberry Pi Foundation

Raspberry Pi Foundation

EDITORIAL

Editor
Meg Wang

Contributing Editor
Dominick Sanders

Subeditors
Louise Richmond, Gemma Coleman

Subscriptions
Dan Ladbrook

Social Media
Sean Sayers

DESIGN

criticalmedia.co.uk

Designer
Dougal Matthews

Photography
The Raspberry Pi Foundation, Adobe Stock

Cover
© MUTI

CONTRIBUTORS

James Abela, Rehana Al-Soltane, Cassandra Allen, Raymond Anacaya, Amy Austin Renshaw, Oliver Brotherhood, Mary Burns, Justin Cannady, Michael Conterio, Aaron Connor, Rik Cross, Paul Curzon, Sofía De Jesús, Shreepriya Dogra, Helen Gardner, Tracy Gardner, Ben Garside, Shuchi Grover, Ben Hall, Joshua Hans, Lisa Hauser, Sam Illingworth, Tiffany Jones, Laura Kirsop, Smita Kolhatkar, Jonty Leese, Alex Luciano, Divya Mahadevan, Sean McManus, Lorena Molina-Castro, Catherine Moore, Kaye North, Alan O'Donohoe, Amanda O'Mara, Marcela Ortiz-Cardona, Beatriz Padín, Mark Peart, Romina Polo, Elena Poncela, Pongprapan Pongsophon, Becca Rose, Ruhee Shah, Kaylla Torres, Jane Waite, Bobby Whyte, Rebecca Winthrop, Julie York

Supported by ORACLE

Contributing Partner
bcs The Chartered Institute for IT Academy of Computing
CAS

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

Hello World is published by the Raspberry Pi Foundation, 37 Hills Road, Cambridge, CB2 1NT. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to skills, products, or services referred to in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Unported (CC BY-NC-SA 4.0).

recycle

(HW) CONTENTS

16–41 COVER FEATURES

CRITICAL THINKING IN THE AGE OF AI
Supporting the next generation as they learn to navigate the challenges and opportunities presented by AI

NEWS, FEATURES, AND OPINION

6 NEWS: CURRICULUM
Unpacking the US National Academies report

8 NEWS: IN PROGRESS
Principles for building education technology

10 NEWS: WHAT WE'RE READING
Introducing the learning sciences

11 NEWS: THE WORD LIST
Concepts you'll read about in this issue

14 EPISTEMIC PLANE
Understanding student programming journeys

16 THE FUTURE OF AI IN EDUCATION
A global report on the use of GenAI in children's education

18 WHOSE KNOWLEDGE IS SHAPING AI?
LLMs are WEIRD — Western, Educated, Industrialised, Rich, and Democratic

20 THE REWRITTEN CSTA PK-12 STANDARDS
A framework featuring core CS concepts interwoven with four pillars of practice

22 PEDAGOGY BEFORE PROMPTS
Why AI professional learning should build teachers' critical thinking

24 AI THROUGH A SOCIOTECHNICAL LENS
What educators bring to the table when teaching with digital technologies

26 STOP DETECTING. START TEACHING.
Tools to teach critical thinking

28 CRITICAL THINKING
Building the skepticism and evaluation skills needed to master AI

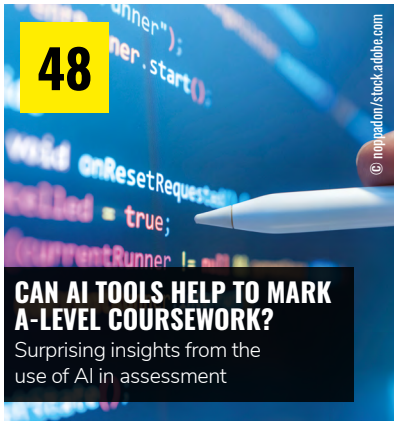
30 BEYOND AI ANSWERS
Helping rural students to develop critical thinking skills for thoughtful AI use

32 KEEPING HUMANS CENTRAL IN AI
Keeping humans in the AI loop and the need for logic in education

34 AI-SUPPORTED APP DEVELOPMENT
What do teachers learn about AI limitations when they become app builders?

12
EVERY CHILD ACHIEVING AND THRIVING
How can every child flourish in our computing classrooms?

36
TURNING STRUGGLE INTO STRENGTH
When students become game designers



48
CAN AI TOOLS HELP TO MARK A-LEVEL COURSEWORK?
Surprising insights from the use of AI in assessment

38 **BUILDING A MULTI-AI SYSTEM**
From single prompts to a multi-AI workflow

40 **DON'T GIVE ME THE ANSWER!**
The role of feedback literacy in strategic AI use

42 **WHERE PHYSICS MEETS TECHNOLOGY**
Transforming the analogue world into zeros and ones

44 **FOSTERING DEEP UNDERSTANDING**
Strategies to help students actively select, organise, and integrate knowledge

46 **FROM DATA TO DECISIONS**
Using micro:bit, Python, and ML to create predictive classroom systems

52 **FROM THREE LIVES TO INFINITE LOOPS**
What the evolution of play means for young coders

54 **COOLEST PROJECTS UK AND IRELAND**
Celebrating the judges' favourite digital creations made by young people at our annual showcase

56 **MANAGING COGNITIVE LOAD FOR DEEPER THINKING**
Using scaffolding to help students organise ideas and communicate clearly

LEARNING

RESOURCES & LESSON PLANS

58 **AI AROUND YOU**
Explore how AI is used in everyday life

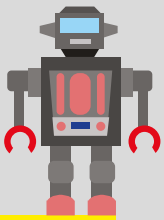
61 **REIMAGINING AP COMPUTER SCIENCE A**
Applied Processing for Creative Scripting and Algorithms

64 **FLATGAMES**
Making games inspired by zines

66 **LEARNING TO TYPE WITH JOKES**
HaHaKeys: a free tool to help primary-age children learn to type

68 **YOUR GUIDE TO PD REQUIREMENTS FOR CS EDUCATORS**
Why professional development matters

70 **BIOBOTS IN COLOMBIA PROGRAMA**
Teaching computational thinking without computers



CONVERSATION

73 **BEBRAS**
A fun computational thinking challenge

74 **MAKING STEM OUTREACH WORK IN BUSY SCHOOLS**
Practical tips on how to reduce friction in STEM outreach

76 **EVOLVING COMPUTING**
How to teach the important skill of decision-making to young people

78 **REDEFINING WHAT A CODER LOOKS LIKE**
Code/Art inspires girls to explore CS careers using art and creative competition

80 **MEET THE CODE CLUB**
How a Code Club partner in Tunisia champions creativity

82 **THE AI OVERLOAD**
Is it all a bit too much?

FOUNDATIONAL COMPETENCIES FOR A DATA- AND COMPUTING-RICH FUTURE

Unpacking the US National Academies report

Shuchi Grover

I had the privilege of serving on the 16-member committee that contributed to the US National Academies report titled 'Data and Computing in K–12 Education: Foundational Competencies' (helloworld.cc/computing-competencies). This National Science Foundation-funded consensus study was two years in the making and the report is very forward-looking. It views data and computing as foundational to all STEM learning. It also responds to the new reality that computing and data increasingly shape civic life and the workforce, yet opportunities to participate in learning in these areas are unevenly distributed and lack coherence across grades and subjects. It's worth underscoring that AI and machine learning rest on a foundation of data and computing, making this work all the more timely.

The problem the report addresses

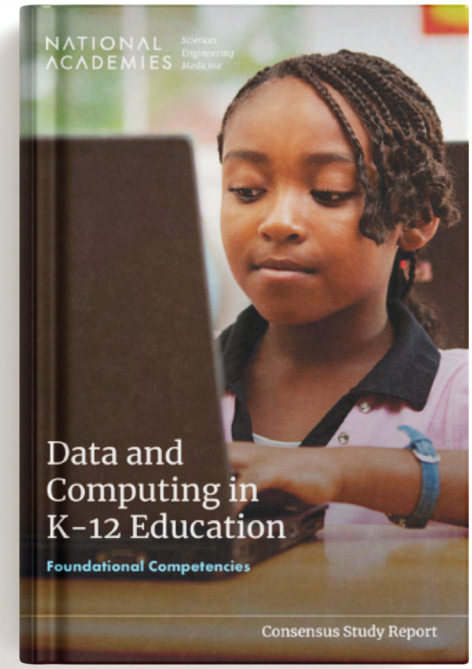
In response to a rapidly changing world infused with big data and an ever-growing, increasingly powerful computational presence, multiple independent groups (advocates for computer science, data science, AI, and quantum information science) are each pushing to add their own content to K–12 education. The curriculum is already stretched thin, and without coordination these well-intentioned efforts risk overwhelming schools, teachers, and students. The report calls this the "messy garden" problem — "ad hoc initiatives that have successfully raised awareness but lack the alignment needed for lasting, equitable impact".

Rather than carving out entirely new curricular territory for each emerging field, the report identifies the common ground that underpins all of them. Three key takeaways frame the report's vision. First, there is a real risk of overload. Simultaneous calls to add computer science, data science, AI, and quantum information science (and the next thing that comes along) to K–12 could overwhelm the system if not approached carefully and coherently. Second, common ground already exists: shared competencies underpin all of these fields, and to some degree they are already embedded in the K–12 curriculum. Third, the report charts multiple paths to better learning — (a) make existing data and computing connections in the curriculum more explicit, (b) enhance those connections within science and mathematics, and (c) expand data and computing in K–12 learning experiences, starting in kindergarten.

Seven foundational competencies

At the heart of the report are seven foundational data and computing competencies that every K–12 learner should develop. These were distilled inductively from an analysis of existing standards and frameworks in computer science, data science, mathematics, statistics, science, and engineering, as well as some top-down deliberation about what learners need.

1. **Problem posing and problem-solving processes:** students learn to define problems, identify steps to address them, attempt solutions using appropriate tools, reflect on the process, and iterate. Problems don't come pre-defined; students develop the ability to formulate questions that data and computing can address.
2. **Producing and working with data:** data is actively produced, not simply collected. Students learn that every measurement involves decisions about what and how to measure, that data provenance matters (who collected it, when, why, and who was left out), and that organising and wrangling data is essential before meaningful analysis can begin.
3. **Abstraction, algorithmic thinking, and automation:** students develop the ability to focus on essential features while filtering out irrelevant detail, design step-by-step solution processes, and recognise when automation is needed as problems scale up. You will recognise this set of competencies as being most closely related to 'computational thinking' — the composite set of skills that shapes many CS education and STEM+CS integration efforts in schools.
4. **Probabilistic and inferential reasoning:**



variability and uncertainty are everywhere. Students learn to identify their sources, develop probabilistic understanding, carry out statistical investigations, and make evidence-based inferences with appropriate degrees of certainty.

5. **Models and representations:** students construct and reason with models to explore phenomena and solve problems, choosing appropriate representations for the data and situation at hand while recognising the inherent uncertainty and limitations of any model.
6. **Technology and society:** students recognise, anticipate, and address tensions related to technology, values, ethics, and societal responsibility — including the practical implications and potential consequences of new and emerging technologies such as AI. This skill is increasingly important as new probabilistic (rather than deterministic) computing paradigms undergird machine learning and generative AI model training.
7. **Data and computing systems:** students develop awareness of how computing tools, platforms, and systems work together to help solve complex problems at scale. Students learn that no single tool is best for all tasks — they learn to choose the right system for their purpose.

You may find that these competencies are not new or exotic — they are recognisable, but they are often taught implicitly or without intentionality. They connect to, extend, and enhance what students already do across the curriculum. Current frameworks for mathematics and science already have many connections with these competencies; this report calls for making those connections explicit and intentional.

Beyond describing the competencies, the report grounds its vision in the science of how students learn — that learning is active, social, and built on prior knowledge and experience. It outlines practical design considerations alongside promising classroom examples spanning elementary, middle, and high school. Crucially, it argues that data and computing are best integrated into existing science and mathematics courses rather than bolted on as separate offerings, and shows through grade-band and subject-specific examples how thoughtful integration can deepen, rather than crowd out, the subjects students already study.

Key recommendations

The report offers 14 recommendations organised into three categories, and they provide a concrete road map for action.

In terms of implementation, the report urges state and local education agencies to build data and computing into existing school subjects in a coherent manner — enhancing, not displacing, foundational mathematics and science. Integration should begin in kindergarten and progress through 12th grade, with standalone courses available as options, particularly in high school. Curriculum developers are urged to use the foundational competencies as a guide for designing learning experiences that include both unplugged (hands-on, screen-free) and plugged (coding and digital) activities, that engage students in meaningful project-based work, and that incorporate ethical reflection on the societal impacts of data and computing.

When it comes to supporting teachers, the report is candid about the challenges and current gaps. Instructional guidance for integrating data and computing is lacking, particularly for elementary teachers. Many teacher candidates graduate without the skills needed to facilitate this kind of instruction. The report recommends tailored professional development within the disciplines teachers actually teach, expanded preservice pathways in computing and data (including partnerships with computer science and data science departments), and ensuring that all preservice teachers — regardless of grade level or subject — develop familiarity with the foundational competencies.

The report calls for a transformation of the current system, including measuring student participation and outcomes, coordinating efforts across professional organisations, and investing in curriculum development, professional learning, and equitable technology access. Notably, it recommends that professional societies in mathematics, science, statistics, and computing revisit their K–12 frameworks to elevate these foundational competencies and signal the urgent need for coherence. It also recommends involving parents and communities in these efforts.

Relevance to the classroom

This report provides crucial pointers and guidance on the integration of computer science, AI, and data science into disciplinary settings. In addition to shared topics and concepts, it offers a shared vocabulary grounded in what we know about how students learn and what competencies will serve them across these fields and throughout their lives. The report also shares several rich examples of integration at various grade levels drawn mostly from research projects in these last few years. The National Academies reports have historically had a significant influence on policy, research, and practice in the USA. We will be watching with interest to see how this latest report is acted upon. (HW)

© wavebreak3/stock.adobe.com

Education technology should support human relationships in the classroom, not replace them

BUILDING EDUCATION TECHNOLOGY YOU CAN TRUST

Putting educators and learners at the heart of our design

Laura Kirsop and Divya Mahadevan

Few people see both the promise and the risks of classroom technology as clearly as computing educators. We teach young people how the technology around them actually works, and we're often the people colleagues turn to when a new tool arrives. That makes us well placed to judge whether the tools we are handed are good quality, and whether they are safe.

That perspective matters at the Raspberry Pi Foundation, where our products are used by millions of young people and educators around the world. Building them safely and responsibly is intrinsic to who we are and what we stand for.

A growing body of research and public commentary links the use of social media, smartphones, and algorithmic platforms to declining mental health and attention in young people. Public attitudes are shifting in response, and parents, educators, and governments are increasingly demanding action. Phone bans in schools and restrictions on social media for young people are spreading across many countries. The technology sector can no longer take public trust for granted.

Generative AI is also raising hard questions for all of us. We believe it should support the relationships that lie at the heart of learning, not replace them. As UNESCO and others have argued (**helloworld.cc/unesco-teachers-ai**), AI in education should support educators, not stand in for them. We agree.

Standing on existing research

We have always been a research-led organisation, and children's digital rights is a rich and well-established field. The United Nations set out a clear framework in 2021 with 'General comment No. 25' (**helloworld.cc/comment-25**), which describes how children's rights apply to the digital environment. Organisations like UNICEF (**helloworld.cc/ritec**) and the 5Rights Foundation (**helloworld.cc/5rights**) have turned that thinking into practical guidance

for the people who build products. Children are not the only ones affected; educators are often asked to use tools without much insight into how they collect data, or how they make decisions about learners. Keeping educators informed and in control is part of designing technology responsibly.

> PUBLIC ATTITUDES TOWARDS TECHNOLOGY ARE SHIFTING AND THE TECH SECTOR CAN NO LONGER TAKE PUBLIC TRUST FOR GRANTED

Eight principles to guide our work

Drawing on this work, we have written eight principles to guide how we build and run our products.

1. **The best interests of the child come first.** We design for children's well-being, which means more than safety and privacy. It includes their agency, their emotional health, their relationships, and the space to create. Sometimes that means choosing against growth, engagement, or speed.
2. **Our technology supports human relationships.** It does not replace them. Any personalised or AI-supported features are there to strengthen the work you do with your learners, not do it for you. People stay in control, and we do not hand decisions about a child's learning to a machine.
3. **We only collect the data we need.** By the time a child turns 13, more than 72 million pieces of personal data will have been collected about them (**helloworld.cc/oaic-exposure**). We collect the minimum we need to help children learn and to measure our impact. We do not sell data, and we never will.
4. **We design for learning, and we prove its impact.** Our products are grounded in evidence about how children learn, and we check whether they achieve what we set out to do. We avoid designs that keep learners hooked rather than helping them learn.
5. **We design with children and educators, not for them.** You are the expert on your classroom. We involve educators and young people in research, testing, and feedback, and we make sure what you tell us shapes the product, not just a report at the end.
6. **We design for diverse needs, abilities, and contexts.** Our technology adapts to people, not the other way around. We do not assume constant connectivity, confidence with technologies, or a classroom that looks like anyone else's.
7. **We are transparent and accountable.** We use plain language to explain how our products work, how data is used, and what rights you and your learners have, and we give clear ways to question or challenge the decisions our systems make.
8. **We apply the highest standards of children's rights everywhere we work.** Where local rules fall short of international best practice, we choose the higher standard. A child's rights should not depend on where they happen to live.

Children's rights apply to the digital environment too

© Drobot-Dean/stock.adobe.com

Committed, not perfect

Publishing these principles is a statement of intent, not a claim that we have got everything right. Regulation is changing fast, the research keeps developing, and our own products grow and change. There are areas where we know we have work to do, and almost certainly areas we have not yet spotted.

These principles are how we hold ourselves to account, and they are only a start. Alongside them, we are mapping the regulatory landscape across the countries where we work, developing ways to assess our products against these principles, and strengthening how we govern this work.

Educators are central to this. You see how these tools land in real classrooms, and you notice things we cannot see from where we sit. If you have a view on these principles, or on the technology you use with your learners, we want to hear it. Tell us where we have got it right, where we have not, and what we have missed. Honest challenge is genuinely useful to us.

Read the full set of principles and sign up to help shape this work at **helloworld.cc/edtech-principles**. (HW)

INTRODUCING THE LEARNING SCIENCES: INSIGHTS AND IMPLICATIONS FOR TEACHING

Bobby Whyte, research scientist from the Raspberry Pi Foundation, shares what he's reading now

Bobby Whyte

INFO

By Shaaron E. Ainsworth , Pryce R. Davis, Peter J. Woods, Jonathan G. Halls, Lenka Schnaubert
Publisher Routledge
Price $42.99
ISBN 9781032029498
URL helloworld.cc/learning-sciences-book

In education, it can be challenging to stay up to date on the current trends in research and consider how they might apply to practice. A new book attempts to resolve this issue by offering educators a practical and accessible introduction to the learning sciences, to help educators better understand the field and how it might inform their work.

But what is the learning sciences? It is an interdisciplinary field that focuses on the study of learning, how it occurs in practice, and how we can design for learning. As the field is interdisciplinary, it is composed of researchers with backgrounds in, among others, education, psychology, computer science, cognitive science, anthropology, sociology, and neuroscience. What makes the learning sciences particularly exciting for educators and resource developers is the commitment to understanding learning in all contexts, both formal and non-formal, and using that understanding to design engaging and interactive experiences to support learning.

Introducing the
LEARNING SCIENCES
Insights and Implications for Teaching
Shaaron E. Ainsworth, Pryce R. Davis, Peter J. Woods, Jonathan G. Halls, and Lenka Schnaubert

What's the book about?

The book looks at teaching and learning through five different perspectives: learning as (i) cognitive, (ii) constructive, (iii) sociocultural, (iv) emotional and motivational, and (v) representational. Each chapter looks at relevant research and its implications and key takeaways for practice. Across five chapters, the authors outline the key mechanisms of learning using the metaphor of architecture to describe teaching.

How is architecture a metaphor for teaching?

As educators, we use many metaphors in the classroom, such as learning as constructing and education as a journey. This book proposes the metaphor of teaching as architecture. Why architecture? In short, we already use architectural language to describe learning processes: learners 'construct' their own understanding and 'build' foundational skills, while educators provide appropriate 'scaffolding' to support learners, and so on. This metaphor is now extended to think about aspects of architecture that have particular applications in teaching: as buildings shape and are shaped by their neighbourhoods, so too is teaching shaped by the communities of practice in which teachers participate.

Teaching is also inherently creative. Like architects, teachers can imbue learning environments with their values, culture, and creativity. There are still rules to follow — structures that are prescriptive — but within those structures, educators can still be imaginative in how they design interactive learning experiences. Likewise, just as we can have an emotional response to a building — I was recently mesmerised by Casa Batlló by Gaudí in Barcelona — so too can innovative teaching invoke an emotional response that motivates and inspires.

Finally, we need tools to construct buildings. Some tools are better than others, and understanding which ones are needed can better support our evolving architecture. Similarly, teaching and learning are representational and require an understanding of representations (for example, diagrams, text, gestures, and simulations) and how they serve as powerful tools for learning.

Who might benefit from this book?

As learning is fundamental to being human, this book will appeal to a wide variety of readers. For teachers, curriculum developers, and teacher educators, it serves as a practical guide to support innovative teaching and learning. Unlike some teaching resources, this book does not offer model lesson plans, recipes for classroom success, nor principles to be adhered to. Rather than restrict readers in this way, the authors outline several key mechanisms and theories to empower readers to apply and adapt these insights to shape their practice. (HW)

NOTEWORTHY TERMINOLOGY

Computer science education is constantly evolving, and so is the language we use to describe this vibrant discipline. Here are ten words you'll come across in this issue, contextualised through the lenses of our contributing writers. Do you agree with these definitions? How are these concepts appearing in your work? Get in touch!

CRITICAL THINKING IN THE AGE OF AI

Spotlight on terms that appear in this issue of Hello World

1. **AI slop:** low-quality, unoriginal, or nonsensical content generated by AI that lacks human intention, depth, or accuracy (pages 32–33).

2. **Cognitive offloading:** a phenomenon where students use tools (like AI) to outsource their thinking, problem-solving, and creativity, potentially weakening their mastery of basic knowledge (pages 40–41).

3. **Ethical design (in gaming):** the practice of creating software that respects a user's time and well-being — for example, building intentional pause points into code to help with self-regulation in a world of 'infinite loops' (pages 52–53).

4. **Epistemic plane:** a research framework used to analyse student programming journeys by comparing their understanding of a concept ('what') with their ability to apply different methods ('how') (pages 14–15).

5. **Feedback literacy:** the ability of a student to interpret, judge, and act upon information (digital, verbal, or written) to improve their own work rather than just passively receiving it (pages 40–41).

6. **Frictionless flow:** a design goal in modern technology and games that removes all difficulty or struggle, which can accidentally skip over the independent problem-solving required for deep learning (pages 52–53).

7. **Hallucinations:** confidently presented but fabricated misinformation generated by large language models, such as fake academic references or nonsensical suggestions (pages 18–19, 28–29, 32–33).

8. **Human-centred design:** an approach that ensures human needs, values, and judgement remain the primary focus when creating or using technology, rather than letting the technology dictate the process (pages 24–25, 34–35).

9. **Sociotechnical approach:** a perspective that views technology not as a neutral tool, but as something deeply connected to social contexts, the priorities of organisations, and human values (pages 24–25).

10. **WEIRD (Western, educated, industrialised, rich, and democratic):** an acronym used to describe the narrow group of people (roughly 12 percent of the world) that features too heavily in the data used to train major AI models (pages 18–19).

EVERY CHILD ACHIEVING AND THRIVING IN COMPUTING

What does every child fulfilling their full potential and flourishing in our computing classrooms look like, and how do we make that happen?

This article was prompted by two recent events, the publication of a white paper and a conference. The white paper was *Every Child Achieving and Thriving* by England's Department for Education (helloworld.cc/every-child). The document outlines a vision for young people in schools and society, including those with special educational needs and disabilities (SEND). There is a welcome shift in language and tone from the deficit model of special needs to framing the strengths of young people and examining what is required for them to thrive. If we narrow our focus for this article, what does 'every child achieving and thriving' mean in computing in practice?

A broad and rich curriculum

The document outlines its ambition for "a curriculum that is knowledge-rich and broad, inclusive and innovative … ensuring that children's horizons are stretched and their opportunities broadened at every stage". Computing and computer science curricula across the world vary greatly in their breadth and ambition, but we can all agree that young people need to learn about the digital world, from foundational knowledge to the functional application of skills, and develop an awareness of the implications of digital technology. A broad, creative curriculum shouldn't just be available to a narrow number of pupils at a specific age, but to all learners throughout their school career. We know from research that creative projects and meaningful, real-world contexts are key to engaging under-represented groups in computer science, particularly girls. As individuals, we have little control over the overarching content and design of our curriculum, but considering the diversity of the projects we teach and how to make them meaningful for learners is a good starting point.

Catherine Moore leads the Computing & Digital Innovation Centre in Sheffield (**helloworld.cc/learn-sheffield**), supporting schools with computing and the wider implementation of educational technology, including AI. She has over 20 years' experience of working with students with special educational needs and disabilities and their teachers, and is passionate about making computing accessible and inclusive for all learners.

High-quality teaching

For all children to achieve in computing, they need to be able to access the learning in lessons. For a minority, this will require specialist support and adaptation. For the majority, high standards and high-quality teaching are key elements of inclusive practice (**helloworld.cc/mainstream-send**). The good news is that there is an ever-increasing repository of research around school-based computing education, and we know better than ever what works. An excellent starting point is the twelve pedagogy principles from England's National Centre for Computing Education (NCCE, **teachcomputing.org/pedagogy**), including a collection of quick reads and podcasts designed specifically for teachers. Have you tried PRIMM (Predict–Run–Investigate–Modify–Make) yet, for scaffolding learning and providing multiple means of interacting with code? Do you follow the semantic wave when teaching abstract concepts, ensuring that students can make links to concrete, contextualised examples, but can also apply their knowledge in a generalised way? Combining computing-specific pedagogy with more generic approaches,

> WE KNOW BETTER THAN EVER WHAT WORKS IN COMPUTING EDUCATION

such as chunking content, supporting comprehension of instructions, and providing structure and routine in lessons, will help all your learners to access the curriculum and engage with the subject.

Building belonging in computing

The second event that contributed to this article was the NCCE's *I Belong in Computing* conference in March. This was an inspiring day spent with like-minded educators exploring how we can foster a sense of belonging in the subject, regardless of gender, socioeconomic status, ethnicity, or learning needs. The white paper similarly acknowledges that "our very best schools know that children achieve most when they feel safe and connected to their school environment" (**helloworld.cc/every-child**), and this connection is built through fostering a sense of belonging. This works best when embedded in the culture across the school, but there are things we can do in computing to contribute to this.

The metaphor of mirrors, windows, and sliding glass doors (**helloworld.cc/bishop1990**) can be a useful lens through which to look at belonging in the classroom. Firstly, mirrors allow young people to see themselves reflected in what they learn. To highlight a diverse and relevant set of role models, take a look at **teachinglondoncomputing.org/diversity** or enrol in the *I Belong* programme from Teach Computing, **teachcomputing.org/i-belong**. Consider developing culturally relevant computing projects (**helloworld.cc/culturally-relevant-computing**). Learners could design digital artefacts in their home language, program a game based on their particular interests, or create a website or poster to celebrate an aspect of their culture or share information about a disability.

Secondly, the windows in the metaphor enable learners to develop empathy and a wider perspective by giving them opportunities to see the world from other viewpoints. You could introduce computing projects that solve real-world problems, look at diverse data sets in data handling projects, or discuss bias in artificial intelligence models. You could teach students explicitly about inclusive design when building websites, apps, or digital content, to ensure they are made accessible to a wide range of users with physical and cognitive needs. For this, Microsoft has tools, guidance, and case studies for older learners: **inclusive.microsoft.design**.

Finally, sliding glass doors allow students to step into and interact with other worlds and reflect on other people's experiences. For example, can students create accessible controllers for gamers with physical disabilities using a Makey Makey (**makeymakey.com**) or micro:bit (**microbit.org**)? Or can students apply web accessibility standards and inclusive design principles to meet the needs of a specific user when planning a digital artefact?

As educators, our aim is always for our young people to thrive and achieve in our classrooms, but it doesn't happen just by wanting it. By delivering broad, inclusive curricula, ensuring we are using high-quality teaching approaches, and actively building a sense of belonging, we can continually develop our inclusive computing classroom with high expectations and opportunities for all our learners. (HW)

#INSIGHTS

BEYOND THE SINGLE LOOP

USING THE EPISTEMIC PLANE TO BETTER UNDERSTAND STUDENT PROGRAMMING JOURNEYS

STORY BY Jane Waite and Paul Curzon

Do you have students who rely on one single programming construct to solve every repetition problem? We call this the single-hammer approach. It's the student who reaches for a simple 'while true' loop for every repetition task, often neglecting other, more suitable constructs. This reliance demonstrates a limited understanding of the computational concept.

Just like a builder with a specialised toolbox, we want our students to understand not only what their program needs to do, but also how to choose the most appropriate, reusable tool for the job. The challenge is to move students from a simple, all-purpose solution to a repertoire of specific, context-appropriate coding patterns. This was the focus of our recent research into student programming constructs, using learning about loops as a case study (**helloworld.cc/epistemic-plane**).

The hidden rules of code: what vs. how

Students writing programs need to understand not only what a program should do, but also how it should accomplish it. This includes essential (but sometimes unspoken) rules of the game, such as the importance of being able to make future code changes easily, or code readability for fixing bugs that are often hidden from novices. These rules change as students progress. While a 'while-true' construct might be intentionally taught as a simple stepping stone to writing working code, better methods must be introduced later to replace this simplistic approach.

To analyse this shift in student practice, our research applied legitimation code theory (**LCT, helloworld.cc/legitimation-code-theory**) by Karl Maton (**helloworld.cc/sociology-possibility**), a sociological framework that surfaces the underlying principles of what is considered legitimate practice. We focused on one analytical tool within LCT: the epistemic plane. This framework analyses the basis of legitimacy for knowledge practice using two continua (see **Figure 1**).

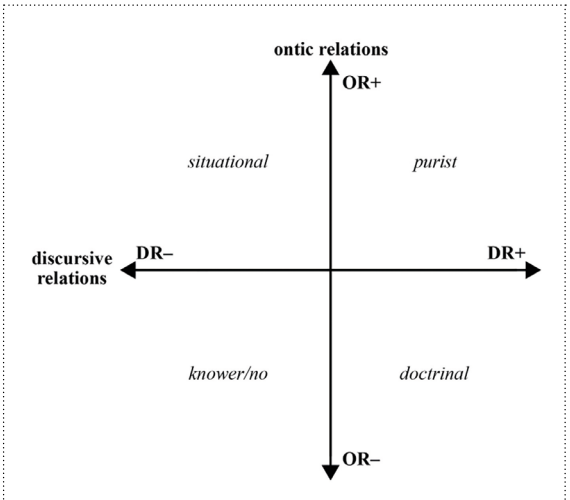


Figure 1 The epistemic plane from **helloworld.cc/epistemic-insights**

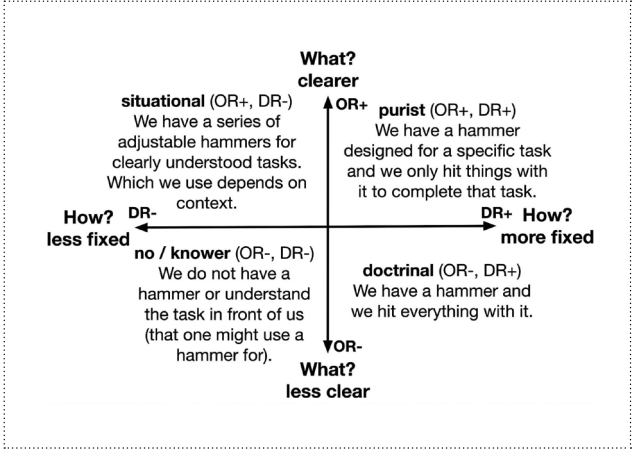
- **Ontic relations (OR): the 'what'**: this axis refers to how clear or well-defined the underlying (here computational) concept is to the student. For example, OR+ means the student clearly understands counter-controlled repetition; OR– means the concept is less clear.
- **Discursive relations (DR): the 'how'**: this axis maps how fixed the method (here programming construct) applied to the phenomenon is. DR+ means the method is fixed (e.g. uses a single programming construct); DR– means there is greater flexibility in the choice of construct.

Understanding student practice: the four insights

The four quadrants, or insights, of the epistemic plane reveal different forms of student practice, highlighted through the hammer analogy (see **Figures 2** and **3**):

- **Purist insight (OR+, DR+):**
  - **The practice**: the concept is clear and the method is fixed. The student recognises a specific computational problem (e.g. counter-controlled

Figure 2 Using the hammer analogy to explain the epistemic plane insights


What?
clearer
OR+
situational (OR+, DR-)
We have a series of adjustable hammers for clearly understood tasks. Which we use depends on context.
purist (OR+, DR+)
We have a hammer designed for a specific task and we only hit things with it to complete that task.
How? less fixed
DR-
DR+
How? more fixed
no / knower (OR-, DR-)
We do not have a hammer or understand the task in front of us (that one might use a hammer for).
doctrinal (OR-, DR+)
We have a hammer and we hit everything with it.
OR-
What? less clear

repetition) and applies the taught, prescribed solution (e.g. a 'for' loop).
- **Analogy**: students have a specialist hammer for a given clear and specific problem and use that hammer for the job.

- **Doctrinal insight (OR–, DR+):**
  - **The practice**: the concept is less clear, but the method applied is fixed. The student uses one fixed approach for all problems that superficially resemble problems they think their tools are for. A weaker student might use a 'while-true' construct as a standard approach for any problem requiring repetition.
  - **Analogy**: this is the single-hammer approach of hitting everything with one tool. This may have been explicitly taught as a simple staging post in lower levels of education.
- **No insight (OR–, DR–):**
  - **The practice**: neither the concept nor the method is clear to the student.
  - **Analogy:** students have no hammer and no clarity of the problem under focus.
- **Situational insight (OR+, DR–):**
  - **The practice:** the concept is clear but the appropriate approach depends on the situation, requiring a choice between methods. The learner understands the problem and selects the most appropriate construct based on the context.
  - **Analogy**: students could adapt a variety of hammers to the job at hand.

Scaffolding the learning journey

The epistemic plane is a tool for analysing learning journeys or the pathways students take over time.

A common journey involves a spiral:

1. **No insight:** novices start here, not knowing what repetition is or how to code it.
2. **Doctrinal insight:** they are introduced to one implementation (like a 'forever' loop in Scratch) and apply it to any repetition problem without understanding the subtleties.
3. **Purist insight:** they learn about specific concepts, like counter-controlled repetition, and the appropriate, clear implementation, such as a 'for' loop.
4. **Situational insight**: over time, learners encounter new loop situations, adapt patterns, and learn to choose between different repetition constructs based on the context.

Teachers play a crucial role in scaffolding these journeys. They must explicitly decide the intended path, which might involve introducing doctrinal solutions as stepping stones. Critically, teachers must explicitly flag when the rules of the game change. Without explicit direction, students may stick to their easier, doctrinal solution and be resistant to change. While situational insight demonstrates flexibility, in software engineering, a standardised, clear (purist) solution is often necessary for teamwork, reuse, and maintainability.

Practical uses for the classroom

The epistemic plane is a valuable, research-informed analytical framework that offers concrete opportunities for educators:

- **Clarifying content and progression**: use the theory to establish and agree the rules of the game at each stage and make them explicit to students. Map out progression to clarify stepping stones and agree on curriculum end points for key concepts.
- **Informing pedagogy**: design your teaching to explicitly guide students toward purist solutions for different specific situations where they are appropriate (counter-controlled repetition, sentinel-controlled repetition, etc.) and explain why and when these solutions are more appropriate than a general doctrinal solution. Once students have a toolkit of purist solutions, design teaching to help them determine when situational solutions are appropriate.
- **Reviewing assessment and misconceptions**: the plane can be used for formative assessment, allowing you to plot student pathways and reflect on misconceptions. It helps surface issues about what kind of solutions are considered appropriate in different contexts, and how solutions act as stepping stones in concept development.

In conclusion, the epistemic plane helps surface the importance of concept development, being aware of end points, and the conceptual shifts that occur, ultimately leading to more transparent and effective teaching and curriculum design.

How do you approach programming problems in your classroom? Do your students rely on one hammer, or are they learning to use a specialist toolbox? (HW)

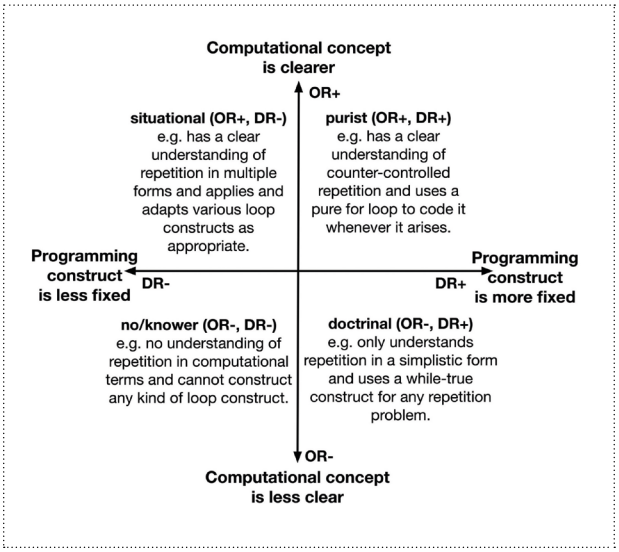

Computational concept is clearer
OR+
situational (OR+, DR-)
e.g. has a clear understanding of repetition in multiple forms and applies and adapts various loop constructs as appropriate.
purist (OR+, DR+)
e.g. has a clear understanding of counter-controlled repetition and uses a pure for loop to code it whenever it arises.
Programming construct is less fixed
DR-
DR+
Programming construct is more fixed
no/knower (OR-, DR-)
e.g. no understanding of repetition in computational terms and cannot construct any kind of loop construct.
doctrinal (OR-, DR+)
e.g. only understands repetition in a simplistic form and uses a while-true construct for any repetition problem.
OR-
Computational concept is less clear

Figure 3 The epistemic plane for learning about repetition when programming

THE FUTURE OF AI IN EDUCATION IS IN OUR HANDS

A new report from the Brookings Institution on GenAI in children's education

What role will artificial intelligence play in shaping the future of student learning and development? Will it substantially improve children's education, or present risks that undermine it? As educators contemplate the integration of AI into classroom practice, how can we embrace its transformational potential while minimising risks to student agency, deep learning, and emotional well-being? And as children increasingly encounter AI everywhere, how do we analyse its impact across all learning contexts, not just within school walls?

These are questions that the Brookings Institution's Center for Universal Education has been investigating since September 2024. Our efforts to provide answers — including holding interviews with hundreds of educators, parents, and students; consulting with education leaders and technologists; examining more than 400 research articles; and hosting a Delphi panel — have culminated in a new report: 'A new direction for students in an AI world: Prosper, prepare, protect' (**helloworld.cc/prosper-prepare-protect**). The report aims to help readers understand the current landscape of benefits and risks around the use of generative AI in children's education.

In the report, we provide a global snapshot drawing on data from 50 countries. It is not intended to be the last word on generative AI and students' learning and development, but the opening of a conversation that poses the question: Are we on the right track?

AI promises greater consistency in education

The benefits of AI in education extend to both teachers and students. By reducing the time spent on numerous teaching-related tasks, AI allows teachers to focus on giving students individualised attention and enhancing the curriculum and instruction. AI can empower student learning by providing access to otherwise unavailable learning opportunities and by presenting content in ways that are more engaging and accessible, particularly for students with disabilities, neurodivergent learners, and multilingual learners.

Through on-demand access in education systems with adequate infrastructure and technology, AI can provide personalised learning pathways, immediate feedback, sophisticated tutoring support, and unprecedented access to educational resources, particularly in under-resourced communities facing teacher shortages. Taken together, these benefits promise greater consistency in quality across education systems.

AI's risks currently overshadow its benefits

The threats of AI tools to students — including the threat posed by unsupervised out-of-school use — are primarily cognitive, emotional, and social. These risks are qualitatively different from the challenges posed by previous educational technologies.

AI tools prioritise speed and engagement over learning and well-being. AI generates hallucinations — confidently presented misinformation — and performs inconsistently across tasks, in what researchers describe as "a jagged and unpredictable frontier" of capabilities (**helloworld.cc/hbs-insight**). This unreliability makes verification both necessary and extraordinarily difficult.

AI's ease of use and its reinforcing outcomes (improved grades with little effort), combined with human tendencies towards shortcuts and the transactional nature of schooling (completing assignments for grades), drive cognitive offloading and dependency, atrophying students' learning — particularly their mastery of foundational knowledge and critical thinking. Young learners lacking this foundational knowledge remain especially vulnerable to accepting AI-generated misinformation as fact. These patterns weaken learning mindsets, as students develop unrealistic expectations about learning ease, lack opportunities to develop resilience and grit, and become less willing to engage in the productive struggles that lead to authentic learning.

The anthropomorphic design of AI platforms make children and youth susceptible to AI's "banal deception" (**helloworld.cc/natale2021**). Its conversational tone, emulated empathy, and carefully designed communication patterns cause many young people to confuse the algorithmic with the human. This conflation directly short-circuits children's developing capacity to navigate authentic social relationships and assess trustworthiness, which are foundational competencies for learning and development.

And in a global community already driven by mistrust of experts, institutions, and one another, AI threatens to fray the relational bonds that are so essential to education. This erosion of trust can provoke cynicism and nihilism in students, undermining the relationships upon which meaningful education depends. Many teachers distrust the authenticity of

student work, while students increasingly question whether their teachers' materials and feedback are genuinely their own. These doubts can extend to fundamental questions about the purpose of education. Does algorithmic information carry more weight than human expertise? Are teachers credible guides when AI can provide answers instantly? Why engage with classmates when AI offers frictionless information? Do educational systems retain value when knowledge can be generated instantly by algorithms?

The evolution of the use of AI in education is in the hands of individuals and institutions. Technology companies, governments, education systems, civil society, teachers, parents, and students all play a role in mitigating AI's risks and harnessing its benefits. We all have to be active participants and ethical stewards, not spectators or bystanders, of AI's impact on children's learning and development.

> LEARNERS AND THEIR TEACHERS CAN PROSPER THROUGH CAREFULLY TITRATED AI USE

Prosper, prepare, and protect

In our report, we argue that action is urgently needed around three interconnected pillars — prosper, prepare, and protect. Together, these pillars support a comprehensive framework of twelve recommended actions for shifting the current trajectory of AI implementation.

Prosper: Learners and their teachers can prosper through carefully titrated AI use (knowing when to teach with and without AI, and using it only when it enhances rather than replaces student effort and cognitive engagement), high-quality pedagogical integration (combining AI with evidence-based practices that prioritise deeper learning), and collaborative design and research (co-designing tools with educators and communities while conducting rigorous research on when and how AI supports learning).

Prepare: The education community can prepare for a rapidly evolving AI-infused world through holistic AI literacy (developing understanding about AI's capabilities, limitations, and implications), robust professional development (equipping educators with knowledge and skills to teach with and about AI), and systemic planning and access (establishing clear visions for ethical AI use while expanding equitable access).

Protect: Technology companies, governments, education systems, educators, and parents can protect students through ethical and trustworthy AI design (protections embedded during the design phase), responsible governance (strong regulatory frameworks), and adult guidance (modelling healthy technology use at home and in schools).

The relative newness of the use of AI tools in education presents opportunities for protecting children from for-profit technology companies whose business models focus on student data and engagement. Educational use cases have not yet become entrenched, creating a window where it remains possible to be responsive to safety and privacy needs. Recognising this, many policymakers are seizing this moment. The European Union's AI Act (**helloworld.cc/eu-ai-act**) employs a risk-based approach that bans unacceptable threats, mandates transparency for limited-risk systems, and regulates high-risk applications, while requiring protections for users' personal information and enforcing age limits for adult-oriented AI. In the United States, where 80 percent of adults support AI safety regulations even if they slow development (**helloworld.cc/ai-safety-gallup**), 31 states have published guidance or policies for AI in K–12 education as of December 2025 (**helloworld.cc/ai-edu-states**). While these policies vary, they often focus on such priorities as online safety, data privacy, and academic integrity.

This means that all of us, individuals and institutions alike, still have an opportunity to shape the future. As our report (**helloworld.cc/prosper-prepare-protect**) makes clear, the impact of AI in education on our most vulnerable citizens — our children — will be determined by the choices we collectively make. These choices must prioritise educational experiences that help all children flourish academically, socially, and emotionally. (HW)

The original version of this article previously appeared on The Brookings Institution website. © 2026 The Brookings Institution. All rights reserved. This article is not covered by the Creative Commons. licence applied to the rest of this magazine.

MARY BURNS

Mary is a consultant for the Brookings Global Task Force on AI in Education. She is an internationally recognised expert in teacher professional development, online learning, and educational technology.

REBECCA WINTHROP

Rebecca is a senior fellow and director of the Center for Universal Education at the Brookings Institution. Her research focuses on education globally, with special attention to the skills young people need to thrive in work, life, and as constructive citizens.

WEIRD IN, WEIRD OUT

Western, educated, industrialised, rich, and democratic (WEIRD): whose knowledge is shaping AI?

It's hard to believe that less than four years ago, most of us had never even heard of large language models (LLMs) such as ChatGPT or Google Gemini. Fast-forward to 2026, and we now have terms such as 'vibe coding' and 'AI slop' appearing in dictionaries.

Teaching is one of the most creative professions, and that creativity has been reflected in the many innovative ways in which LLMs are already being used in education, from experimenting with AI tools to help draft lesson plans or adapting activities for different learners, to reducing some of the repetitive workload that comes with teaching. As these tools become more common in education, it is worth asking whose knowledge, language, and perspectives these systems are built from.

Data provenance

Imagine a history teacher planning a lesson on empire or migration. A lesson built from one textbook or national perspective may tell a very different story from another that draws on multiple voices, cultures, biases, and interpretations. Teachers are used to thinking critically about where data comes from, whose perspectives are represented, and how source selection shapes understanding.

LLMs may seem very different, but they raise similar questions at a much larger scale. Their responses are shaped by enormous training data sets gathered from across the internet, from websites, books, forums, social media platforms, videos, and increasingly even AI-generated content itself. Researchers refer to this as data provenance: understanding where AI training data comes from and how those origins influence the outputs we receive.

A recent large-scale audit of nearly 4000 AI data sets found that modern AI systems increasingly rely on web crawlers, social media, and synthetic data sources, with platforms such as YouTube becoming dominant sources for speech and video training data.

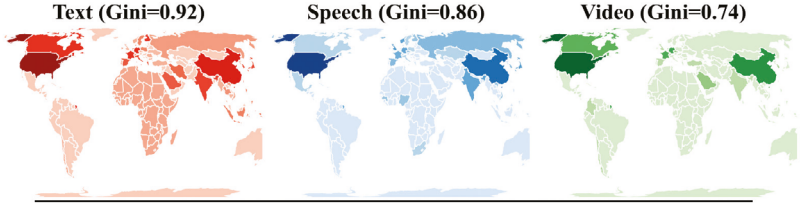
In the paper 'Bridging the Data Provenance Gap Across Text, Speech and Video' (**helloworld.cc/longpre**) we find that "Despite some differences in European, Russian, and Middle Eastern representation, creators are heavily concentrated in the US, China, and Western Europe, with little to no representation in South America or Africa, across modalities." This means that the output of an LLM will reflect the views, cultures, languages, assumptions, and priorities that are most heavily represented in its training data, and those not represented will be absent.

Are we all WEIRD?

In another paper I read recently, 'Which Humans?' (**helloworld.cc/atari**), I came across the acronym WEIRD: Western, educated, industrialised, rich, and democratic. Researchers originally used the term to highlight a problem in psychology research (**helloworld.cc/henrich**). Much of what has historically been described as normal human behaviour was actually based on the study of a relatively small part of the world. Surprisingly, only around 12 percent of the world's population are considered WEIRD. Despite this, WEIRD societies have often been treated as the default, and this idea becomes especially important when thinking about AI in education.

LLMs are trained on huge amounts of human-created data, but that data is not evenly sourced from across the world. The 'Which Humans?' paper found that ChatGPT's responses most closely resembled cultures such as the US, Canada, the UK, the Netherlands, and Australia, while being far less aligned with countries such as Ethiopia, Pakistan, and Kyrgyzstan.

At first glance, this might seem like a language issue. Perhaps an AI model is



	Africa	Asia	Europe	N America	Oceania	S America
			By Count			
Text	0.3	13.4	24.0	61.5	0.7	0.2
Speech	3.6	35.7	30.4	30.4	0.0	0.0
Video	0.0	25.2	24.4	48.0	0.8	1.6
			By Tokens or Hours			
Text	0.0	6.1	55.4	38.4	0.1	0.0
Speech	0.1	38.8	18.8	42.4	0.0	0.0
Video	0.0	23.1	22.0	38.2	16.7	0.1

Figure 3 by Longpre et al., from the paper 'Bridging the Data Provenance Gap Across Text, Speech and Video' (**helloworld.cc/longpre**) used under CC BY 4.0

■ Data set creators are heavily concentrated in the US, China, and Western Europe, with little to no representation in South America or Africa

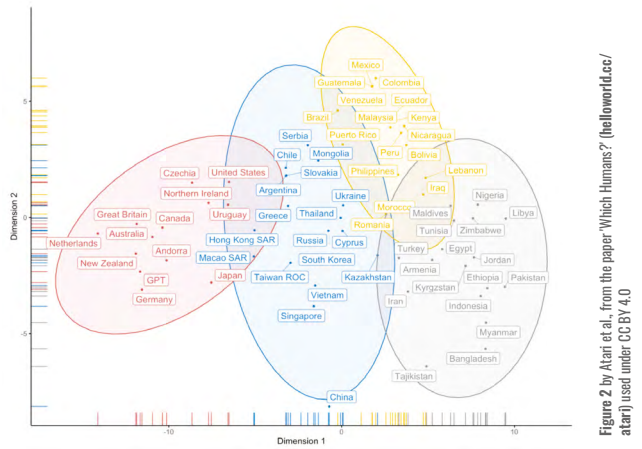


Figure 2 by Atari et al., from the paper 'Which Humans?' (helloworld.cc/atari) used under CC BY 4.0

■ A visual map illustrating the psychological distance between various global cultures and the AI model; groups and nations on the left are those whose values and thinking styles align most closely with the model

less accurate when writing in non-WEIRD languages, or less capable of understanding local dialects. But the influence goes much deeper than spelling or translation.

I recently read Lucy Crehan's *Cleverlands* (**helloworld.cc/crehan**), which explores education systems around the world, and one example particularly stayed with me. In Japan, schools place a strong emphasis on collective responsibility. Crehan describes the Japanese proverb 'The nail that sticks out will be hammered down' as reflecting the importance many Japanese schools place on group harmony and social cohesion. In contrast, many Western education systems often encourage students to stand out, challenge ideas, and express themselves individually. Neither approach is universally correct, but the contrast reveals how deeply culture shapes ideas about behaviour, success, and learning.

The same applies to AI systems. A model trained predominantly on WEIRD data may absorb assumptions that many users never notice because they just feel normal. Ideas about what makes a good student, how professionalism should be expressed, or whether people describe themselves through personal achievements or through family and community relationships can vary significantly across cultures.

One of the most interesting findings from the paper was that ChatGPT tended to reflect more individualistic Western styles of thinking. In practice, this means AI-generated content may reinforce particular cultural norms while presenting them as universal. In a classroom context, students may not only be learning computing concepts or practising programming tasks, but also absorbing hidden assumptions about identity, communication, and belonging that are embedded within the examples and explanations AI systems generate.

If this all sounds a bit abstract, these cultural assumptions become much easier to spot when language models are asked to generate stories about people.

A recent Raspberry Pi Foundation computing education research seminar featured Dr Thema Monroe-White presenting her paper 'Intersectional biases in narratives produced by open-ended prompting of generative language models' (**helloworld.cc/monroe-white**).

The study asked several major LLMs to generate short open-ended stories about classrooms, workplaces, and relationships. Importantly, the prompts did not explicitly mention race, gender, or ethnicity. Yet the models still produced remarkably consistent patterns in who appeared as the star student, the mentor, the professional, or the person needing help.

In one example, models were asked to generate stories about a star student mentoring a struggling student in a language class. White-associated names such as Sarah and John appeared in the star student role, while names associated with Latino, Black, Asian, and Middle Eastern identities appeared far more often as the struggling student. One generated story described John mentoring Maria, a student "who had recently moved from Mexico", in a Spanish language class, despite Spanish being her native language. Another portrayed Ahmed as an engineering student dealing with "language barriers" while Charlie patiently guided him to success.

None of these prompts asked the models to stereotype anyone, but the patterns emerged anyway. Developers did not intend for these outcomes to be produced. Rather, the models appear to absorb and reproduce patterns that are already embedded in the data they are trained on. When AI systems are trained on large-scale cultural patterns from the internet, they do not simply reproduce language; they also reproduce assumptions about power, competence, belonging, and identity.

Implications for teaching critical thinking

Many of us are approaching AI tools with a mixture of curiosity, caution, and creativity. We are not looking to automate teaching; we are looking for practical ways to support learners, reduce workload, and spend more time focusing on the parts of teaching that matter most.

What I hope this article surfaces is that these systems are built from human data and are shaped by particular cultures, languages, histories, and power structures. What an AI model presents as neutral, helpful, or normal may actually reflect a much narrower slice of humanity than we realise. If GenAI increasingly shapes the examples, explanations, and stories our students encounter, then learning to question those assumptions may become one of the most important forms of critical thinking we teach. (HW)

BEN GARSIDE

Ben is senior learning manager for AI literacy at the Raspberry Pi Foundation. A former teacher, he leads content for Experience AI, a global AI literacy programme providing free resources for educators.

CS EDUCATION IN THE AGE OF AI

Teaching students to question, not just execute

During a network analysis lab, a cybersecurity student proudly showed an instructor a script he had generated with AI. The code ran perfectly and produced the expected results. But when the instructor asked a few follow-up questions regarding what a specific line of code actually did, where the data was coming from, and what would happen if the script ran on a production network, the room grew quiet. The student paused. The script worked, but he could not fully explain why. In cybersecurity, that pause matters.

That moment captures a growing reality in computer science (CS) classrooms. AI can generate answers instantly, but understanding and evaluating answers still requires human judgement. In a world moving at supersonic speed, CS education needs to find a new balance. Students must learn to think critically so they can direct AI rather than being directed by it. This was one of the driving factors behind the newly rewritten CSTA PK–12 Standards (**csteachers.org/pk12standards**), a research-backed framework designed to prepare students for a future that is arriving faster every day.

To translate these high-level goals into classroom reality, the development process focused on diverse perspectives and educator expertise. The revised standards team was composed of PK–12 educators and members of the broader CS education community to ensure that the standards reflected authentic classroom practice while remaining grounded in research. The team carefully examined implementation, cross-referenced related standards, and kept students and teachers at the centre of every decision. Throughout the process, the team applied the dispositions already embedded in the standards (curiosity, persistence, critical thinking, and so on) to the complex work of drafting and revising.

Teachers have already built strong practice around core CS concepts and student-centred pedagogy, and this revision builds on this foundation rather than starting from scratch. The goal was not to render existing resources obsolete, but to strengthen and extend them. Teachers will recognise the familiar concepts of Algorithms & Design, Programming, Data & Analysis, Systems & Security, and Computing & Society (see **Figure 1**). These core concepts support the development of fundamental dispositions: creativity, critical thinking, curiosity, persistence, reflectiveness, resourcefulness, and sense of belonging. AI content is distributed throughout the five concepts, and additional new priorities such as the societal impacts of computing are intentionally interwoven through four sets of practices (Ethics & Social Responsibility, Human Centred Design, Computational Thinking, and Inclusive Collaboration). Each standard was carefully constructed to incorporate a concept, one or two practices, and one or two dispositions.

Drawing from the higher levels of Bloom's Taxonomy (**helloworld.cc/blooms-taxonomy**) for computing, the updated standards were crafted to promote deep critical thinking. For example, in second grade (ages 7 to 8), students are asked to create an algorithm that includes sequence, events, and iteration to solve a problem or express ideas. The use of 'create' signals a shift toward higher-order thinking, even at early grade levels. The standard is supported by a boundary statement that clarifies what the student is expected and not expected to do in the process of mastering the standard, ensuring students are learning developmentally appropriate concepts and excluding the use of complex industry-focused tools. The clarity around each of these standards will help teachers implement them accurately.

Developing a clear progression across grade bands was another key priority. The team carefully considered how standards build from early childhood through high school, ensuring concepts are neither redundant nor disconnected. This progression helps teachers understand their own grade-level expectations in relation to other levels and enhances

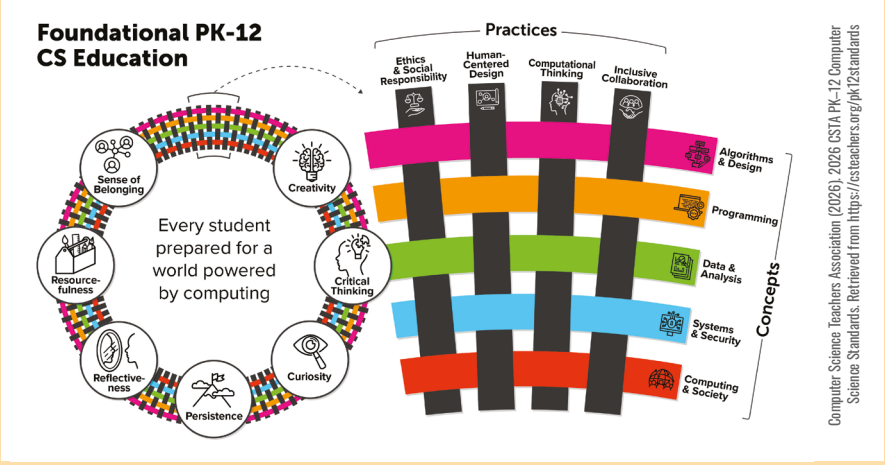


Figure 1 The 2026 CSTA PK–12 Standards interweave core CS concepts with practices and support the development of key dispositions

students' critical thinking across ages 4 to 18. Let us look at how machine learning progresses as an example:

- In elementary school, students begin by identifying patterns and sorting information. They may explore how a computer recognises similarities, while also noticing when it makes mistakes. Students are encouraged to question every step: Why did the system group these items together? What might it be missing?
- In middle school, students begin working with data sets and simple models. They test systems, analyse outputs, and start identifying inconsistencies. Students might notice that a model works well in some cases but fails in others, prompting questions about bias, accuracy, and missing data.
- By high school, students analyse and critique machine learning systems more deeply. They evaluate outputs, identify limitations, and consider broader impacts. In cybersecurity contexts, this may include questioning whether an AI-generated script is secure, or whether a model could be manipulated or misused.

> STUDENTS NEED TO LEARN TO THINK CRITICALLY SO THEY CAN DIRECT AI INSTEAD OF BEING DIRECTED BY IT

Across all grade levels, the focus remains consistent: students are not only building systems, they are learning to question them.

Interdisciplinary connections were also a critical focus. The standards are designed to support integration across content areas, making computing more accessible to all teachers. A second-grade student might create an algorithm for a story in English Language Arts. A middle-school student might analyse data in science. A high-school student might explore bias and fairness in social studies. These connections reinforce that computing is applicable across disciplines and that CS education extends far beyond career preparation.

The standards include both plugged and unplugged examples that can be applied in a variety of classroom settings. These examples also reinforce the practice of inclusive collaboration and computational thinking. In one activity, students work in pairs or small groups to design an algorithm for a simple dance using cards or visual coding blocks. They then exchange their algorithms with another group to test for clarity and completeness.

Critical thinking was not just a goal of the 2026 CSTA PK–12 Standards; it shaped the process used to develop them. Although developed in the US, the standards can serve as a reference for global educators and schools, as the concepts covered are essential for all students. Built on a strong foundation and with access to guidance, examples, and professional learning, teachers will be well supported in bringing these standards to life.

As emerging technologies continue to transform the world around us, computing education must evolve as well. In an era when AI can generate answers instantly, teaching students to question, analyse, and think critically about those answers may be one of the most important lessons we can provide.

The student in the cybersecurity lab could generate and run the script, but could not explain it. That gap between execution and understanding is exactly where CS education has to live now. Knowing how to run a tool is no longer enough; students need to know why it works, when to trust it, and what to do when it fails. With the 2026 CSTA PK–12 Standards as a foundation, the next generation of students won't just run the script, they will know how and why it works, and perhaps most importantly, will be able to grapple with the ethical implications of the technologies they build. (HW)

AMANDA O'MARA

Amanda is the K–6 STEM teacher at Gilmour Academy's Lower School in Cleveland and an at-large board member of CSTA Ohio. She has been teaching CS and STEM at K–8 grades in public, rural, and independent STEM-designated schools for ten years (**linkedin.com/in/amandajomara**).

SMITA KOLHATKAR

Smita is the assistant head for innovation, responsible AI, and edtech at Gideon Hausner Jewish Day School, California, and president of the CSTA Silicon Valley chapter. She has taught CS and AI to elementary and middle-school students for about two decades (**linkedin.com/in/skolhatkar**).

TIFFANY JONES

Tiffany is an engineer-turned-educator who is now a high-school computer science and cybersecurity teacher at Global Impact Academy STEM Magnet High School in Georgia. She has been a full-time educator for six years and currently serves as a CSTA Georgia chapter leader (**linkedin.com/in/tjonesatl**).

PEDAGOGY BEFORE PROMPTS

Justin Cannady and **Shreepriya Dogra** share why AI professional learning should build teachers' critical thinking, not just their confidence in using tools

Generative AI is moving quickly through education, and teachers know it. They are seeing it show up in lesson planning, student writing, and classroom conversations about what counts as learning. Yet much of the professional development being offered to educators still starts in the wrong place. It focuses on what button to click, what prompt to write, or which platform to try next.

That approach may create a quick burst of excitement, but it does not prepare teachers for the decisions that matter. Teachers do not just need help with using AI tools. They need support with thinking critically about when AI should be used, why it should be used, and what is gained or lost when it enters a learning experience. They need space to ask harder questions: What does this tool ask students to do cognitively? Does it support learning, or shortcut it? What new barriers, biases, or privacy concerns might it introduce? How do we stay human amid all this?

Those questions shaped the design of our introductory AI workshop for K–12 educators. We did not want to build training that treated teachers like passive recipients of the latest edtech trend; we wanted to create a professional learning experience that positioned them as thoughtful instructional decision-makers. In other words, we wanted AI professional development to strengthen professional judgement, not replace it.

Teachers collaborating during the introductory AI workshop where they were focused on pedagogy, ethics, and classroom decision-making

Purpose over proficiency

Our workshop was built across three days, with sessions moving from foundational ideas about AI, to classroom applications, to the design of more inclusive and future-ready learning experiences. We used the TPACK framework (**helloworld.cc/tpack**) to help educators think about the relationship between technology, pedagogy, and content knowledge, rather than treating AI as a standalone topic.

This mattered, because many educators were not asking for another isolated tool demo; instead, they were asking how AI fits into the real work of teaching. They wanted to know how to evaluate a tool in relation to a learning goal, how to make sense of AI within their subject area, and how to avoid using AI tools in ways that flatten student thinking.

So instead of organising the workshop around 'best AI tools for teachers', we organised it around questions of purpose. Participants explored core AI ideas, discussed ethical issues, and examined how AI tools might support or interfere with good teaching. They worked with tools, yes, but tool use was never the main event. The goal was not proficiency for proficiency's sake; the goal was deeper reasoning.

This shift is important. When professional development focuses too heavily on tool fluency, teachers can leave with a handful of tricks but little confidence in adapting, critiquing, or making principled choices. AI changes too fast for a tool-only approach to hold up anyway. The platform of the month will come and go; the need for critical thinking will not.

If we want educators to integrate AI tools well, we have to help them ask better questions: What kind of thinking is this task meant to develop? Where should

students struggle productively, and where might support be appropriate? Is AI being used to enhance feedback, generate ideas, model examples, or bypass the learning altogether? These are pedagogical questions before they are technical ones.

Ethics before efficiency

One of the most important decisions we made was refusing to treat ethics as an add-on. Too often, ethics appears at the end of an AI workshop like an afterthought everyone is expected to nod at before moving on. Bias, privacy, transparency, and accountability are mentioned, but they are not meaningfully integrated into the design work. That sends the wrong message: it suggests ethics is separate from practice, when in reality it is part of every instructional decision.

We wanted educators to experience something different. Across the workshop, participants were invited to think about AI as a human-centred issue, not just a technical one. That meant discussing what it means to remain the decision-maker in the room. It meant examining how AI systems can reproduce bias or obscure how answers are generated. It meant recognising that convenience is not the same as quality, and efficiency is not the same as learning.

For teachers, this kind of ethical reasoning is not abstract. It shows up in ordinary classroom moments. Should a student use an AI tool to revise a paragraph? Should a teacher use AI to generate feedback? Should a class use an AI image generator for a history task? The answer is rarely a simple yes or no. It depends on the learning goal, the context, the students, and the responsibilities attached to the decision.

This is where critical thinking becomes essential. Educators need more than permission or prohibition; they need frameworks for judgement.

In our experience, teachers responded well to this. Many were relieved to engage with AI tools in a way that honoured the complexity of their work. They did not want to be told to either embrace AI uncritically or reject it entirely; they wanted room to think, question, test, and reflect. Frankly, that is the kind of professional respect teachers deserve.

Agency through intention

Another lesson from this work is that one-time training is not enough.

Participants appreciated having a structured introduction to AI, but they also made it clear that sustained support matters. Teachers wanted continued opportunities to collaborate, revisit ideas, and adapt what they learnt to their own contexts. That should not surprise us. Good teaching is contextual, and meaningful professional growth takes time.

If schools and systems are serious about AI literacy, they need to stop treating professional development as a rapid-fire orientation to tools and start treating it as an ongoing opportunity to build teacher agency. That means making space for experimentation, discussion, and revision. It means differentiating support. It means helping educators connect AI not just to technology standards, but to disciplinary goals, assessment practices, and the realities of their students' lives.

> WE HAVE TO HELP EDUCATORS ASK BETTER QUESTIONS WHEN INTEGRATING AI TOOLS

It also means being honest: some uses of AI tools may support critical thinking, and some may weaken it. Not every shiny use case deserves classroom time. Teachers need permission to slow down and decide carefully.

For us, this is the larger takeaway. In the age of AI, we not only want our students to develop critical thinking; we also want teachers to practise it actively. They must evaluate claims, question outputs, consider consequences, and make informed instructional decisions in fast-changing conditions. Professional learning should help them do exactly that.

So, where should educators begin? Not with the fanciest tool, and not with a list of prompts. Begin instead with the learning. Begin with the human beings in front of you. Begin with the question, 'What kind of thinking do I want this experience to develop?' AI may be new, but that question is not. It has always been at the heart of good teaching. (HW)

JUSTIN CANNADY

Justin is a learning experience designer at the University of Minnesota's Northern Lights Collaborative for Computing Education. He designs inclusive CS and AI professional learning for K–12 educators and students (**jcannady@umn.edu**).

SHREEPRIYA DOGRA

Shreepriya is a PhD student in learning technologies at the University of Minnesota, Twin Cities. She explores how emerging technologies, particularly generative AI, can be reimagined in teaching and learning through participation, agency, and equity (**dogra013@umn.edu**).

AI THROUGH A SOCIOTECHNICAL LENS

What educators bring to the table when teaching with digital technologies

Much of the current research about artificial intelligence in education has focused on how young people and students demonstrate their learning and how they can learn to use AI tools responsibly, as well as how educators can detect and deter AI-generated work to safeguard academic integrity, and how current curriculum and assessment practices may be feeling the pressure from generative AI tools. These are all legitimate and important concerns, but they risk overlooking a critical part of the picture: we, ourselves, as educators.

A sociotechnical approach

Educational technology has long been understood as more than a collection of tools used in neutral or predictable ways. Drawing on the sociotechnical approach, scholars such as Selwyn (**helloworld.cc/selwyn**) have argued that technologies do not simply arrive in classrooms and fix an issue or improve practice by default. Instead, they form part of an interrelated, symbiotic relationship with social contexts, institutional priorities, power relations, policy pressures, and personal and professional values. A sociotechnical approach reminds us that technologies are always entangled with human judgement and decision-making. What matters is not just what a tool can do, but how it is integrated with pedagogy, who uses it, under what conditions, for what purpose, and in what context.

This perspective is particularly important as AI technologies begin to feature more prominently in educational settings. Much of the current debate focuses on the capabilities of AI: its speed, fluency, or apparent intelligence, and its downfalls and limitations. Framing this through a sociotechnical lens shifts our attention elsewhere. It also invites us to ask how AI is understood by teachers, how institutional policies frame its use, and how existing beliefs about technology, learning, teaching, and assessment shape how AI is taken up or rejected. The aim of this article is not to advocate nor to condemn AI, but to acknowledge that our beliefs about digital technologies and AI, whether we see them as threats, shortcuts, partners, or problems to be managed, matter just as much as the tools themselves in determining how they are used, resisted, or reimagined in teaching and learning.

The role of educator assumptions

Teachers, and all users for that matter, do not encounter AI as neutral observers. We bring our own assumptions, values, professional identities, and experiences into our classrooms. How we feel about AI, whether we experience it as a risk to academic integrity, a source of discomfort, a curiosity, or a potential collaborator, shapes the role it is allowed or not allowed to play in practice. Before asking what AI means for learners, it is therefore worth asking what it means for us as educators.

A sociotechnical perspective reminds us that how educators understand AI matters just as much as the technology itself. Selwyn and others have long argued that educational technologies succeed, fail, or mutate not because of their technical qualities alone, but because of how they are interpreted and enacted by people in particular settings. Technologies are filtered through what educators believe about learning, teaching, assessment, authority, and effort. These beliefs are rarely neutral, and they are often implicit.

Research into teachers' beliefs about technology suggests that these assumptions act as powerful mediators between access and use. For example, some authors highlight that educators' beliefs about what counts as 'good teaching' often determine whether technologies are used in transformative ways or simply absorbed into existing routines. When digital tools align with teachers' existing values and pedagogical commitments, they are more likely to be adopted in meaningful ways; when they challenge those commitments, resistance is not only likely, but rational. This places teacher knowledge firmly at the centre of any discussion about AI in education. It requires educators to be clear about what they teach, why they teach it, how learning is understood to occur, and the role technology might reasonably and ethically

FURTHER READING

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 319–340. **(helloworld.cc/davis)**

Rogers, E. M. (2003). *Diffusion of innovations* (5th ed.). Free Press. **(helloworld.cc/rogers)**

Selwyn, N. (2021). *Education and technology: Key issues and debates* (3rd ed.). Bloomsbury Publishing. **(helloworld.cc/selwyn)**

Educators respond to new technologies differently: some experiment, others wait and observe, some resist until external pressures force change

play within that space. This can help explain why the same AI tool can appear as a creative support in one classroom and an unacceptable risk in another. For an educator who values learning as a process of exploration and dialogue, AI might be framed as a thinking partner or a prompt for critique. For another, committed to learning as individual demonstration of mastery, the same tool may appear to undermine fairness or effort. Neither position is inherently irrational; each reflects deeper assumptions about what education is for.

Educator response profiles

These differences are not new. As Rogers reminds us, individuals and institutions respond to new technologies in patterned ways (**helloworld.cc/rogers**). Within any educational system, some educators are more inclined to experiment, others to wait and observe, and some to resist until external pressures force change. When applied to educational technology, this has often been described in terms of educational technology profiles, ranging from early adopters to more cautious or sceptical users. Importantly, these profiles are not about competence or enthusiasm alone, but about trust, perceived relevance, and alignment with professional identity. AI technologies make these differences more visible, because they touch on particularly sensitive areas of practice: assessment, authorship, academic honesty, and professional authority. As a result, responses to AI are often emotionally charged. What may be framed publicly as a technical or policy issue frequently rests on deeper personal and professional beliefs.

Technology acceptance factors

Another useful way of understanding educators' responses to AI comes from the Technology Acceptance Model (TAM; **helloworld.cc/davis**). This model suggests that people's willingness to adopt a technology is strongly influenced by factors like perceived usefulness and perceived ease of use. While often applied simplistically, TAM highlights something important in this regard. If educators do not believe a technology adds value to learning, or if it is seen as creating additional risk or workload, adoption is unlikely, regardless of how powerful the tool seems to be. When a technology is perceived as pedagogically useful, manageable, and aligned with existing practice, educators are far more likely to explore, adapt, and integrate it into their teaching.

In the context of AI, perceived usefulness is closely tied to beliefs about teaching practice. If AI is seen primarily as a tool that enables shortcutting or surface performance, its usefulness will rightly be questioned. If, however, it is understood as a way of supporting feedback, metacognition, and accessibility, its perceived value changes. Ease of use matters, but not in isolation; tools that are easy to use but hard to justify pedagogically may still be resisted.

MARK PEART

Mark is a lecturer in teacher education (primary education — technologies), at the University of Glasgow in the UK. He is a fellow of the Centre for Transformative Change in Schools and currently serves as the programme lead for the MSc AI in Education.

Professional judgement

Taken together, these perspectives reinforce a central point about educators' engagement with AI. Responses are shaped less by ignorance or fear than by coherent belief systems built over years of practice. A sociotechnical approach does not seek to overcome or circumvent these beliefs, but to surface and examine them. Rather than asking why teachers are for or against AI, a more productive line of enquiry is to also ask what assumptions about education underpin those positions.

This matters because the future of AI in education will not be determined solely by policy documents or platform capabilities, but by the everyday judgements educators make about what is acceptable, valuable, and ethical in their own contexts. Recognising and scrutinising our own assumptions is key to safeguarding our agency as educators, positioning teachers not as passive recipients of AI technologies, but as active, informed, and critical decision-makers. Without this emphasis on agency, there is a risk that decisions about AI use become driven by external mandates or technological possibility, rather than by questions of professional judgement and educational purpose. (HW)

Microsoft Copilot was used to support language editing and proofreading. Responsibility for the final content rests with the author.

For teachers, student AI usage is a given; the priority is now whether students understand its capabilities and limitations

© unai/stock.adobe.com

STOP DETECTING. START TEACHING.

Sam Illingworth looks at why AI detection tools are failing students and shares two free browser games that can help build critical AI literacy

BOT OR NOT

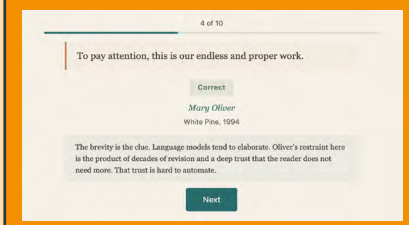
Bot or Not: Test your AI detection skills

Aim: Discover whether you can distinguish human writing from AI writing

Materials: Any device with a browser

1. Go to **helloworld.cc/bot-or-not**.
2. Read each of the six texts carefully.
3. For each text, decide whether it was written by a human or an AI tool.
4. Review your results. Most players score close to 50 percent.
5. Discuss as a class: What clues did you look for? Were they reliable?

Key question for students: If you cannot reliably tell the difference, should we trust a tool that claims it can?


4 of 10
To pay attention, this is our endless and proper work.
Correct
Mary Oliver
White Pine, 1994
The brevity is the clue. Language models tend to elaborate. Oliver's restraint here is the product of decades of revision and a deep trust that the reader does not need more. That trust is hard to automate.
Next

'Bot or Not' tests learners' ability to detect whether a text was written by an AI tool or a human

Most schools have responded to generative AI the same way, by trying to catch students using it. AI detection tools like Turnitin's AI detector and GPTZero promised to solve the problem, but they have not; they have created a new one.

AI detection tools do not reliably distinguish between human and machine writing. A 2023 study from Stanford University found that these tools disproportionately flag the writing of non-native English speakers as AI-generated (**helloworld.cc/liang**). Students who write in a second language tend to use simpler sentence structures and more common vocabulary, and so do large language models — the detectors cannot tell the difference.

This is not a minor error rate; it is a structural bias baked into how the tools work, and the consequences fall on the students least able to challenge them.

The wrong question

The problem is not the technology, but the question we are asking. 'Did a student use AI to write this?' is a question about policing. It assumes the goal is to catch dishonesty. But generative AI is not going away. Students will use it in every job they walk into. The more useful question is: 'Does this student understand what AI can and cannot do?'

That is what critical AI literacy means. Not how to use AI tools, not how to write better prompts, but the ability to evaluate AI outputs, recognise their limitations, understand the systems behind them, and make informed decisions about when to use them and when not to.

What this looks like in a classroom

I have spent the past year building free tools that bring critical AI literacy into classrooms without requiring any technical expertise from teachers.

The starting point is simple: most people cannot tell the difference between human and AI writing. They think they can, but they are wrong, and that overconfidence is exactly why detection tools feel trustworthy when they are not.

'Bot or Not' (**helloworld.cc/bot-or-not**) is a browser game that tests this directly. Players read six short texts and guess whether each one was written by a human

FLAGGED

Flagged: AI detection simulator

Aim: Experience the real consequences of AI detection bias

Materials: Any device with a browser

1. Go to **helloworld.cc/flagged**.
2. You are the AI detection officer at a university. Review each flagged case.
3. Read the student's profile, their writing, and the detector's confidence score.
4. Decide: Is this student cheating, or has the system made an error?
5. After all the cases, review how many students you identified correctly.

Key question for students: Who is most likely to be wrongly flagged, and why?

In the game 'Flagged', players are AI detection officers responsible for reviewing student cases

cheating, or if the system has got it wrong.

The game is built on real patterns from research. Non-native speakers get flagged more often; neurodiverse students who write in atypical patterns get flagged; students who write clearly and simply get flagged. The detector is confident every time, but is often wrong.

Why games work here

These are not gamified quizzes, but experiences designed to produce a specific feeling: uncertainty. The moment a student realises they cannot tell human writing from machine writing, or the moment they wrongly expel a student based on a confident detection score, something shifts. They stop trusting the tool and start thinking about the problem.

That emotional response is harder to achieve with a worksheet or a slide deck. Games create situations where the learner has to make a decision and then live with the result. That is closer to how critical thinking actually develops.

Getting started

Both games are free, run in any browser, and require no setup. A teacher can use them in a single lesson.

Critical AI literacy does not mean banning AI

Both games include shareable results so that students can compare scores and discuss their reasoning.

The shift we need

AI detection is a dead end. The tools do not work reliably, they penalise the wrong students, and they teach nothing. Critical AI literacy is the alternative. It does not mean we have to ban AI or pretend it doesn't exist. It asks students to think carefully about what these systems are, what they produce, and why that matters. That is a skill worth teaching. Detection is not. (HW)

> WHEN STUDENTS REALISE THEY CANNOT DISTINGUISH HUMAN AND AI WRITING, THEY STOP TRUSTING THE TOOL AND START THINKING

or by an AI tool. The results are consistently humbling. Most people perform close to chance. The game makes the case more quickly than any lecture could: if you cannot reliably tell the difference, a detection tool built on the same principle cannot either.

'Flagged' (**helloworld.cc/flagged**) takes this further. Players step into the role of university AI detection officers, reviewing student submissions that have been flagged by a detection tool. Each case includes the student's profile, their writing, the detector's confidence score, and contextual clues. Players must decide if each student is

'Bot or Not' works well as a starter activity. Play it as a class, with students voting on each text before the answer is revealed. The discussion that follows is where the learning happens: Why did you think that one was human? What made you sure? What does 'sure' even mean when the best you can do is guess?

'Flagged' works best as a longer activity or homework. Students play through all the cases individually, then compare their decisions. The debrief is the point: Who did you flag? Why? What would you do differently knowing what you know now?

SAM ILLINGWORTH

Sam is a professor of critical AI literacy at Edinburgh Napier University and the founder of 'Slow AI', a newsletter and curriculum for critical AI literacy with over 12,000 subscribers. His free browser games for AI literacy have been played over 13,000 times (**theslowai.substack.com**).

© Lomb/stock.adobe.com

WHY AI CAN'T REPLACE THE SKILL OF CRITICAL THINKING

Building the skepticism and evaluation skills needed to master AI

FIVE QUICK WINS FOR ANY AI LESSON

1. **The ritual fact-check:** Every lesson, take one AI output and verify it together as a class. Make it normal.
2. **Prompt, then interrogate:** Students write a question and get an AI answer, then identify one thing to challenge.
3. **AI vs. expert:** Compare AI output with a textbook or other reliable source. Discuss what's different and why.
4. **Wrong on purpose:** Ask an AI tool a trick question. Analyse why it fails — and what that tells you about how it works.
5. **Exit reflection:** End with, "What did I actually learn today vs. what did AI just give me?"

Let's be honest: our students are already using AI. They are asking it to explain algorithms, debug code, and — yes — write their homework. The question isn't whether to allow it, but whether we are teaching them to use it well.

That means teaching them to question it, to push back, to recognise when a fluent, confident answer is also a wrong one. In other words, it means teaching critical thinking.

As computing teachers, we're perfectly placed to lead this. Debugging is critical thinking. Evaluating the efficiency of an algorithm is critical thinking. We already do this — we just need to extend it to AI itself.

The confidence problem

The biggest risk AI poses in the classroom isn't laziness but misplaced trust. Large language models produce text that sounds authoritative whether or not it's accurate.

A 2023 study documenting ChatGPT's use in scientific writing found that it confidently fabricated references — inventing paper titles, authors, and journal details that simply didn't exist — with no indication given that anything was wrong (**helloworld.cc/alkaissi**). If it can fool researchers, it can certainly fool our students. We have all been fooled by students — and technology — confidently telling us something is true when we know it is false.

Students and staff who accept AI output without questioning it are not learning — they are outsourcing their thinking. And the danger isn't just academic. It's the habit of mind that forms when you stop asking, "But is this right?"

YOUR CRITICAL-THINKING TOOLKIT

Before you use any AI tool in class, ask:

- [ ] Who built this, and what's their incentive? It is worth understanding the business models of free tools.
- [ ] What could go wrong? Name a plausible failure mode before you start.
- [ ] Where does it sit on the SAMR model (Substitute–Augment–Modify–Redefine)? — be honest.
- [ ] What thinking will students do? If an AI tool does the thinking, redesign the task.
- [ ] How will we verify the output? Make fact-checking part of the lesson, not an afterthought.

> AI CAN SCAFFOLD THE LOWER RUNGS OF BLOOM'S REVISED TAXONOMY, BUT THE UPPER RUNGS REMAIN STUBBORNLY HUMAN

Bloom's Revised Taxonomy is a useful frame here (**helloworld.cc/anderson**). AI can scaffold the lower rungs of remembering, understanding, and applying, but the upper rungs of analysis, evaluation, and creation remain stubbornly human. Our job is to make sure students are climbing those rungs themselves, not hitching a ride.

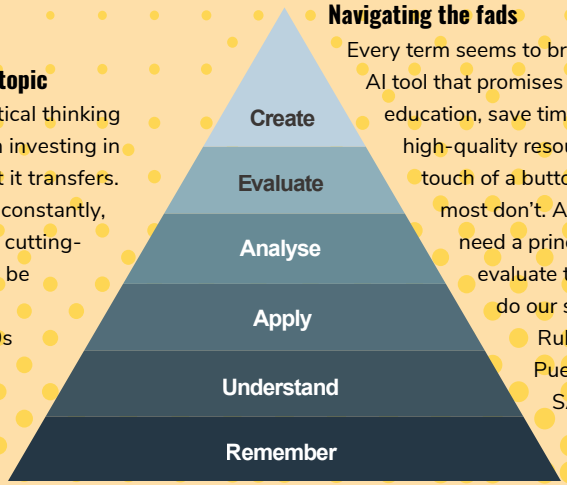
A toolkit, not a topic

What makes critical thinking especially worth investing in right now is that it transfers. AI tools change constantly, and whatever is cutting-edge today may be obsolete by the time your Year 9s sit their GCSEs. Think dabbing (**helloworld.cc/dabbing**), bottle flipping (**helloworld.cc/bottle-flip**), and of course the 6-7 trends (**helloworld.cc/6-7**) that have been and gone in the student hemisphere. Teaching students to think critically about AI isn't about any one tool — it's about building habits of mind that work on whatever comes next.

Researcher Peter Facione (**helloworld.cc/facione**) identified five core critical-thinking skills that hold up across every domain: interpretation, analysis, evaluation, inference, and self-regulation. Map these onto AI use and you get a set of questions every student should be asking automatically:

- **Interpretation:** What is this tool actually doing with my prompt?
- **Analysis:** What assumptions or biases might be baked in?
- **Evaluation:** Is this source credible? Could this be wrong?
- **Inference:** What conclusions can I reasonably draw from this output?
- **Self-regulation:** Am I thinking here, or just accepting?

These are not new questions — they are the questions good thinkers have always asked. AI just makes them urgent again.



Bloom's taxonomy

Navigating the fads

Every term seems to bring a new AI tool that promises to transform education, save time, and create high-quality resources at the touch of a button. Some do; most don't. As teachers, we need a principled way to evaluate them — and so do our students. Ruben Puentedura's SAMR model (**helloworld.cc/puentedura**) is worth keeping handy. Does this tool Substitute something we already do, Augment it slightly, Modify the task meaningfully, or Redefine what's possible? A tool that just automates a task students should be doing themselves sits at the bottom of that ladder. We should apply to AI tools the same critical lens we want students to apply to AI content.

The bigger picture

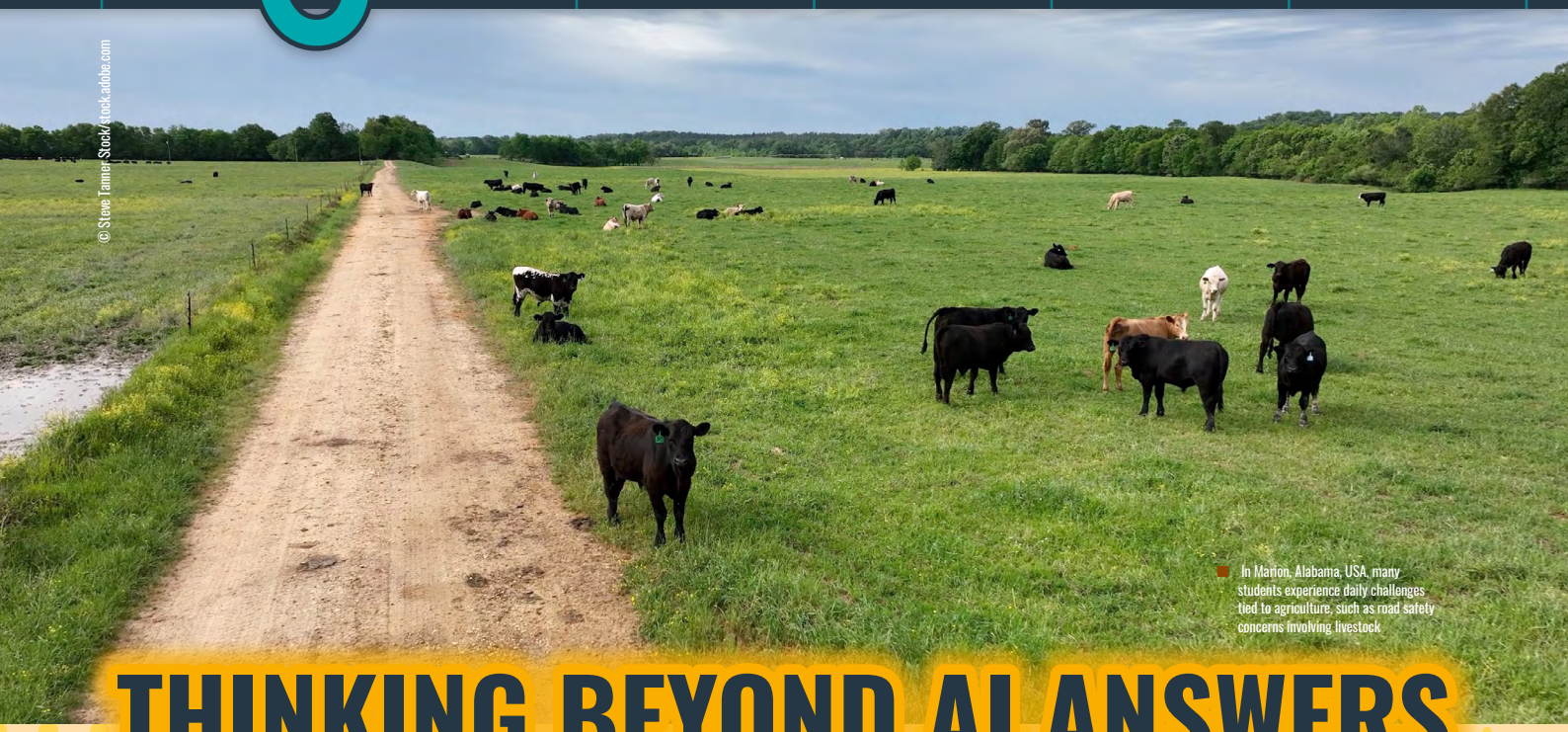
We are living through a genuine shift in what it means to think and learn. AI is not going away, and our students will use it throughout their working lives. The risk isn't that they use it, but that they may use it without judgement.

As computing educators, we have a head start. Our subject already values precision, scepticism, testing, and iteration. We just need to make those values explicit in how we talk about AI — and model them every time we bring a new tool into the room.

The best thing we can give our students isn't a list of approved AI tools, but the confidence to question any tool, including those that don't exist yet. (HW)

JONTY LEESE

Jonty is an associate professor at the Centre for Teacher Education at the University of Warwick in the UK. He is head of the international PGCE programme, specialising in computer science teacher training. He supports trainee primary and secondary teachers locally and internationally, with a strong emphasis on helping educators navigate the digital landscape with confidence, critical thinking, and creativity.



© Steve Tanner Stock/stock.adobe.com

In Marion, Alabama, USA, many students experience daily challenges tied to agriculture, such as road safety concerns involving livestock

THINKING BEYOND AI ANSWERS

Cassandra Allen shares how place-based computer science projects help students in rural Alabama develop critical-thinking skills while learning to use AI thoughtfully

A student once said, 'AI gave me the answer.' I responded, 'But how do you know it's right?'

That moment captures a challenge I see more and more in my classroom. AI tools can generate responses quickly, and students naturally gravitate towards that speed. But my goal as a computer science teacher is not simply to help students get answers — it's to help them think through problems, question results, and make informed decisions.

I teach in Marion, Alabama, in the United States, where many of my students experience daily challenges tied to agriculture, transportation, and infrastructure. Road safety concerns involving wildlife and livestock, changing weather conditions, and limited resources are part of everyday life here. Because of that, I design place-based projects that connect computer science to problems students recognise. When students work on issues that affect their own community, they are less likely to accept AI answers at face value and more likely to ask whether a solution actually works for their environment. In the age of AI, teaching critical thinking has become just as important as teaching coding.

The AI shortcut

Students today have access to tools that promise instant solutions. When AI generates an answer in seconds, it can feel as if the work is already done. I started noticing that students were accepting AI-generated responses without questioning whether the solution made sense or truly solved the problem.

Instead of avoiding AI tools, I decided to teach students how to use them responsibly. We talk openly about what AI does well and where it falls short. I ask students to pause and evaluate results before moving forward: Does this answer make sense? What assumptions is the tool making? What happens if we test it?

These questions shift the focus from speed to understanding. Students begin to realise that AI can provide ideas, but it cannot replace their thinking. Critical thinking starts when students stop asking, 'What's the answer?' and start asking, 'Why does this work?'

Programming builds thinking

One of our most meaningful projects focused on a real problem in our community: car accidents caused by deer and cattle crossing roads. Instead of solving an abstract programming task, students were asked to think about how technology could help improve safety.

At first, students looked for quick solutions, and some turned to AI for ideas. But as they explored those suggestions, they realised that answers that sounded good on screen didn't always make sense in real-life conditions. They had to think about visibility, environment, and practical limitations. The conversations changed from, 'What should we build?' to 'Would this actually work here?'

Programming supported this process because students had to test their ideas. When something didn't work, they couldn't just move on — they had to

investigate why. Debugging became a lesson in patience and logic. Students learnt to revise, test again, and explain their reasoning.

During a mock AP Computer Science Principles performance task, students also discovered that how they communicated with AI tools mattered. I encouraged them to ask for multiple options — sometimes requesting five different solutions — and then evaluate which one best met their goals. This pushed them to compare ideas, justify their decisions, and refine their thinking. They quickly learnt that writing a strong prompt requires the same clarity and precision as writing good code.

© CandyRetriever/stock.adobe.com

Focusing on local issues encourages students to critically evaluate AI solutions and their real-world efficacy

Automation in context

Critical thinking continued as students worked with automation in our classroom growing space. This isn't just a garden — it's a living system that students help to design and manage. They use soil and moisture sensors to monitor conditions, they consider irrigation needs, and they decide how automation should respond to changing data.

Students soon realised that automation introduces new challenges: How do we protect electrical components from rain and heat? How much moisture is enough before irrigation activates? What happens if a sensor gives inaccurate data?

These questions forced students to think beyond simply using technology. They now had to consider systems, inputs, outputs, and real-world constraints. AI tools could suggest possible approaches, but students had to determine what worked best for their specific environment. Through testing, adjusting, and troubleshooting, they learnt that automation depends on thoughtful decision-making.

Over time, I've seen students become more confident in questioning technology. They test ideas before accepting them and approach problems with curiosity instead of looking for the quickest answer.

Supporting thinking

AI is becoming a normal part of education, and it has the potential to support learning in powerful ways, but the most important skill we can teach students is how to think critically about the tools they use. Programming, place-based problem-solving, and automation projects help students move beyond quick answers and towards deeper understanding.

When students stop asking AI tools to think for them and begin using them to support their own thinking, they are learning more than computer science — they are learning how to make thoughtful decisions in a world shaped by technology. (HW)

> "WRITING A STRONG PROMPT TAKES THE SAME PRECISION AS WRITING GOOD CODE

© CineLens/peopleimages.com/stock.adobe.com

Instead of avoiding AI tools, we need to teach students how to use them responsibly

CASSANDRA ALLEN

Cassandra is a computer science and business education teacher at Francis Marion School in rural Perry County, Alabama. She is passionate about expanding access to computer science, AI literacy, and technology education for students in underserved communities.

Don't throw the baby out with the bathwater, and don't throw the AI out with the tech slop

This image was generated by the author using Gemini

KEEPING HUMANS CENTRAL IN AI

Sofía De Jesús shares her take on the importance of keeping humans in the AI loop and the need for logic in education

Change is a reality we can sometimes struggle with. In education, the last 20+ years have brought a barrage of technology applications that have transformed classrooms worldwide and changed how we approach learning. Whether that change is net positive or net negative is something we debate often, and we are now faced with even more changes and considerations, and one critical question: What is the role of technology in my classroom and in my learning?

Who threw the baby out?

The idiom 'Don't throw the baby out with the bathwater' is a warning to avoid throwing away something wanted in the process of getting rid of something unwanted. I sometimes wonder who threw the baby out with the bathwater in order for this saying to exist, but the sentiment is important. Some AI is 'slop' and, indeed, harmful, but some AI tools are extremely helpful. When we are thinking about uses of AI, what is the bathwater and what is the baby?

For me, using AI as a personal therapist is a definite bathwater use. A GenAI system cannot feel or truly understand anything a human says. What it can do is look at the training data, find similar situations, and craft a response based on those situations. There is absolutely no actual thinking involved, just algorithmic instructions. So while the 'advice' or 'therapy' received may sound real and accurate, it is in no way real. Mental health is such a high-stakes part of our well-being; it should not be ruled by algorithmic instructions that don't take human feelings into account. We cannot program feelings; we can program to mimic them, and this is not the same. This GenAI use should go out with the bathwater.

Now for the baby. There are some very helpful uses of AI for thinking through processes, ideas, and planning. Say I'm trying to digest content to prepare for a debate in a class. There are GenAI tools that let me feed in my own sources, such as NotebookLM, to build a repository I can then interact with. I can ask questions about the sources, use the tool to tease

out possible gaps in the literature, and practise arguments with the AI bot. While hallucinations will necessarily still happen (it is impossible to remove hallucinations, or errors, completely from generative outputs), I think this use stays with the baby.

Nurturing the baby

The key to these positive uses of AI is that the human side of things always has to be central to how we talk about AI, how we make decisions about it, how we create policies and guidelines, and how we ensure privacy concerns are alleviated. And it has to be central to how we create with AI, for example writing with it. Many talk about keeping the human in the loop; I prefer to go further and make sure AI is always human-centred. There is no need for AI without the human, and we should never talk about AI or GenAI as if it were a living, breathing being.

Another way of maximising the positives of using AI tools in the classroom is to encourage logical thinking. In education, since the push for educational technology and 21st-century skills, we have been swayed by buzzwords. We talk about critical thinking as if it were a new concept, or project-based learning, which is a rebranding of experiential learning. But all of these concepts are learnt through logic. We used to teach logic in a very intentional way, through early childhood activities in which students select clothing based on weather, or middle- and high-school geometry lessons in which students work through truth tables and proofs.

But over the past two decades, we have gradually removed much of the backbone of that logic. In geometry, we sometimes don't cover proofs in as much depth, or at all, and many textbooks have even removed truth tables. Many teachers do not use textbooks at all, which may lead to less adherence to established goals, objectives, and standards. And I could argue that this dates back to calculator usage, because we have now moved courses that require abstract reasoning to a time well before the child's brain has developed it. This means we have been forced to also remove content they are not ready for, while celebrating the ability to do more work earlier.

Knowing how to get the right answer does not necessarily mean we understand the concepts. We can learn a mathematical algorithm without understanding what it represents. We have removed logic from being woven into our curricula, and we do not teach it explicitly, but it is a requirement for critical thinking. We cannot think critically if we cannot assess premises logically.

> "WE DO NOT TEACH LOGIC EXPLICITLY, BUT IT IS A REQUIREMENT FOR CRITICAL THINKING

Logical reasoning with GenAI

In order to work with GenAI, we have to develop logical reasoning skills. Because the outputs are flawed, we have to assess them against our knowledge base and imagine the outcomes that could come out of its use.

A few years ago, a generative AI summary stated that you could use glue on pizza to keep the cheese from sliding off (**helloworld.cc/ai-pizza-glue**). If we accept the outputs of GenAI without analysing them, we can harm ourselves and our students. I don't think anyone wants to eat glue on their pizza. Now, we hope common sense tells us not to use glue, but if we get used to accepting algorithm outputs without being critical, our senses could dull to the point of thinking, 'Maybe the AI is on to something', leading to glue-eating shenanigans. (Please do not try this at home.)

Embracing change doesn't mean we are not critical of the changes; it means we see the value, but we also see the pitfalls, and we analyse each situation to ensure we are making the best decisions with the technologies we have. It is imperative that, along with those technologies, we emphasise the need for logic, logical reasoning, and logic education in general at all stages in schools. This is a call to intentionally highlight lessons in logic at each grade level and adapt, add, and advance logical reasoning education in order to make better use of AI.

We, the humans, are the creators, the critics, the users, and the deciders of what AI uses we throw out with the bathwater. We, the humans, should be the ones who are moulding the way we use emerging technologies. (HW)

© Videophilia/stock.adobe.com
Logical reasoning skills or common sense tells us that glue is not a good pizza ingredient

SOFÍA DE JESÚS

Sofía is a CS strategy specialist at the University of Texas at Austin, USA. She is an author with over 26 years of experience, which includes teaching, leading teacher training, and working as an editorial director in educational publishing.

AI-SUPPORTED APP DEVELOPMENT

What do teachers learn about AI limitations when they become app builders?

At first, AI was horrible at coding. When I tried to build an app with ChatGPT, the code was always broken, and my colleagues and I confidently agreed that this would never replace human programmers. The more complicated the app was, the worse the results were. But very quickly, things started to change: new AI development platforms launched, code assistants improved, and I began to realise that AI systems were getting much better at building real tools.

Building my first app with AI

Over the last year, I've been exploring how AI tools write code and experimenting with different AI coding tools both in and out of the classroom. I started by thinking about what I needed. The first app I created was a teacher translator that live-transcribed my speech into multiple languages for students in my classroom. It was a tool built purely out of necessity for my own teaching practice (**helloworld.cc/teach-translator**).

Building this teacher tool with AI taught me a huge amount: how AI tools work under the hood, what information websites store, how APIs behave in live settings, and even how pricing and app monetisation decisions are made. It also gave me an amazing opportunity to talk with my students in real time about the decisions involved in app development in the context of a genuine, real-world project.

Exploring AI-supported platforms

After this, I started to explore other AI-supported app development platforms such as Claude, Replit, Base44, and Lovable. I found that each platform had different strengths and weaknesses, and I realised that combining my existing coding and computer science knowledge with these tools made me remarkably effective.

Claude is particularly good at generating Python and HTML apps, and, as long as you have somewhere to host the code, it's a wonderful coding companion. Lovable lets you both build and host apps directly on the site. It prompts you to confirm every significant change, such as database edits, and expects you to build more from scratch. Base44 almost builds the app for you and then lets you tweak from there. It's very beginner-friendly, but like Lovable, it focuses on apps that you can host online.

Working with students

With my students, we built an arcade machine from scratch, and we used AI to guide us through the entire process. We used Claude to learn Python, understand and comment on code, and troubleshoot when things didn't work, including how to fix issues on our Raspberry Pi in the most effective way. It felt like having a personalised Raspberry Pi expert on hand while we built and wired our machine. This AI expert knew what we were doing and kept track of the problems and solutions we had already tried. Claude lowered the barrier so that every student in my classroom could create apps. That's when I realised I could offer the same kind of empowered experience to teachers.

Working with teachers

I took this learning to other teachers and asked: What would happen if teachers could design and build the exact apps they needed? What if they could explore app development, understand the limitations of AI, and experience the kind of problem-solving that computer scientists go through? I've had many conversations with teachers about what student data is collected and stored, what is legal or illegal to collect, how students log in, what APIs are, which ones are necessary, and how much they cost. I've also talked with students and teachers about ethics: what is being built, why it's being built, how it will be used, and how it serves everyone involved.

The second teacher tool I built is a free web app that helps teachers organise groups (**helloworld.cc/group-up**). When I was making it, I collaborated closely with other teachers and students to refine the idea. Teachers set up their classrooms and define collaborative roles, then share a classroom code with students. Students enter the

SOME NIFTY AI CODING SITES

Lovable (lovable.dev): Creates and hosts web apps with strong security controls

Claude (claude.ai): My class go-to for Python, HTML, and other languages; requires external hosting

Base44 (base44.com): Beginner-friendly; builds and hosts web apps

Replit (replit.com): Codes all languages with built-in preview; doesn't host for public use

GitHub Education (github.com/education): Free for teachers/students; excellent tools and classroom management

KEY THINGS TO THINK ABOUT AROUND AI TOOL DEVELOPMENT

Data privacy: What is collected? Is it legal and secure?

Purpose: Who is this for? What does it simplify?

Necessity: Does it already exist? Is it helpful?

Joy factor: Does it save time on tedious tasks?

Safety: Is data protected from misuse?

Ethics: How does it serve all parties? Is it accessible?

Cost: What are the API costs? Is it sustainable?

Hosting: Where is it hosted? How do users connect?

code, add their first name and an emoji, and select their preferred roles. They can also designate one role they never want to take on. The teacher can then decide group sizes and generate groups. They can also specify which students should or shouldn't work together and add late-joining students. This is a highly specific tool that only a teacher is likely to design; it comes from years of playing student Tetris like coaches arranging players on a football team!

Since I started doing this, teachers have begun approaching me with ideas for their own apps. One teacher wanted to build a sports injury tracking app, which led to a rich discussion about the legality and ethics of collecting and storing that kind of sensitive information safely. Another teacher asked about an app that could record what was said in lessons — essentially a dictation tool — and then generate summaries. We ended up exploring existing tools and resources that could do this safely and securely, without sending student or teacher audio to third-party AI services that might mine or reuse the data.

The idea of teachers using AI systems to build their own teaching tools is both epic and empowering. Creating apps and going through the process teaches a lot about computer science and the ethics and limitations of AI. AI can't do anything without us. When you build, debug, and test an app, it becomes very clear how essential human creators are in directing and using the tool. As one of my students put it: "AI is a tool, just like a hammer. It won't build anything by itself. People using hammers build, and teachers are definitely up for the challenge."

A student playing games created by other students on the Raspberry Pi arcade machine built with help from AI

The coolest thing about AI-supported app development is that it allows far more people to participate in creation, especially teachers. As a lifelong crafter, I love the thought that everyone can make a piece of art. The best artists are those who really understand their process, materials, and platform; in this context, that deep understanding often belongs to computer scientists. But by using AI as a coding assistant and supporting teachers to make their own tools, we invite many more people into the conversation and the art form of creating programs. Imagine a world in which every teacher experiences empowered coding; that's a world where every teacher understands that every student deserves that same sense of creative power. (HW)

JULIE YORK

Since 2002, Julie has taught computer science and media at South Portland High School in Maine, USA, creating courses in animation, game design, digital graphics, and more. She centres community, representation, and joy in her teaching and life.

USING AI TOOLS TO TURN STRUGGLE INTO STRENGTH

When students become game designers

In many schools, students who struggle with a skill are positioned as the ones who always need more help and intervention. I wanted my students to experience something different — to see that the very areas they found difficult could become places of leadership, creativity, and purpose. I designed a lesson using Canva AI (helloworld.cc/canva-ai), in which students learnt to write prompts and use AI tools to create games based on topics they struggled with and skills teachers had identified as least mastered. What began as a lesson in prompting and design quickly became something more: an opportunity for students to turn struggle into service.

Starting with what hurts

The skills students find hardest can easily become the places where they lose confidence. A child who struggles with reading or maths can begin to believe that difficulty means inability, and over time that feeling quietly shapes how they see themselves as learners. I wanted to interrupt that story. Instead of hiding from the hard topics, my students faced them directly. We looked at classroom data, teacher recommendations, and the skills that needed the most reinforcement across the school, then asked a different question: What if these were the very topics we used to create something meaningful? Students were no longer simply being told what they had not mastered; they were being invited to build something that could help someone else.

The heart of this lesson was not the tool; it was the purpose. Each student was asked to design a game for other students, which gave the work dignity and told them that their ideas, experiences, and even their struggles had value. They were creating something that could support a peer in learning.

Teaching more than a technology skill

Before they could create a game, students had to learn to write prompts that were specific and intentional. Using Canva AI was not about handing students a shortcut; it was about teaching them to communicate their thinking with clarity and purpose. They learnt that AI tools do not respond well to vague thinking, so they had to describe the skill involved, the audience, the format, the visuals, and the learning goal. In this way, prompting became much more than a technology skill. Students had to decide what they wanted the game to teach, who it was for, and what would make it clear and engaging for another learner. When AI-generated results were weak or misaligned with the goal, students revised their prompts, tested the game, and improved it. They learnt that good design requires patience, reflection, and iteration. They were not just using AI; they were learning how to think more clearly.

That kind of purpose changes the way students work. I watched them become more careful with their word choice, more thoughtful with their questions, and more intentional with visuals, directions, and feedback. They were no longer thinking only about what they liked; they were thinking about what another learner might need. That is empathy in action. For students who have often been on the

Students learnt that good game design requires patience, reflection, and iteration

RAYMOND ANACAYA

Raymond is an interventionist at Olanta Creative Arts and Science Magnet School in South Carolina, with a PhD in educational leadership. He has dedicated his career to improving reading comprehension and numeracy skills through STEM projects. Raymond was named Coastal Pee Dee STEM Educator of the Year 2024 and was one of the Top 5 Finalists for South Carolina.

receiving end of intervention, becoming the person who creates support for others is powerful. It restores agency and builds pride. It reminds them that they are not defined by the skills they struggle with; they are also capable of helping others grow.

Why equity must be at the centre

For me, this lesson is deeply connected to equity. In many under-resourced schools, students do not always experience technology as a space for creativity, authorship, and innovation. Too often, they are asked to consume digital content rather than create it, clicking through programs and responding to systems built by someone else. Rarely are they positioned as designers. Equity is not simply about giving students access to devices or digital tools; it is about giving them meaningful opportunities to use those tools in ways that affirm their intelligence, voice, and potential. It means making sure that students, especially those who are too often underestimated, are trusted with work that requires imagination, problem-solving, and leadership. This project did that. It also centred equity by focusing on least-mastered skills and teacher-recommended topics. We were not creating games just for entertainment; we were creating with intention, based on real academic needs in our school community. That meant student creativity was directly connected to student support, making the project innovative and responsive. It told students that their ideas could help solve real learning challenges.

Beyond digital skills

What moved me most was not simply that students learnt to write prompts or build games, but how they began to see themselves. Students who are sometimes quiet found ways to contribute. Students who struggle in traditional academic settings showed deep insight when thinking about what makes a question confusing or how to make a game more helpful. Students who had once needed support in a skill became the ones designing support for others. There is something beautiful about that kind of reversal — a child who once felt behind now creates a tool to help someone else move forwards, and a student who once doubted their ability now sees their idea come to life. For a learner who may not always feel successful in school, being positioned as a creator, thinker, and contributor can shape identity in powerful ways.

> BEING POSITIONED AS A CREATOR, THINKER, AND CONTRIBUTOR CAN POWERFULLY SHAPE IDENTITY

What I learnt as a teacher

One of the biggest lessons for me was that AI integration becomes far more meaningful when it is grounded in authentic need. The power of Canva AI was not simply in its ability to generate content quickly, but in how it supported student thinking, voice, and agency. I also discovered that students are capable of more sophisticated design thinking than we sometimes assume, especially when the task is relevant and purposeful. When they know their work could help a classmate understand a difficult skill, their motivation and attention to detail change. Their sense of responsibility grows. This lesson reminded me that meaningful technology integration is not about using the newest tool for its own sake, but about using tools to expand access, deepen learning, and position students as capable contributors.

Reimagining what AI can be in schools

There is understandable fear and uncertainty around AI in education. Many educators worry that it will make students passive, reduce effort, or replace genuine thinking. This experience reminded me that AI tools can also be used differently. They can become bridges for expression, tools for design, spaces where students practise revision, communication, and critical thought, and a means for young people to create something meaningful for their community. In our classroom, Canva AI did not replace thinking; it demanded it. Students had to imagine, plan, revise, analyse, and empathise. They had to make decisions that mattered and create with others in mind. Most importantly, they had to believe that what they made could help someone else.

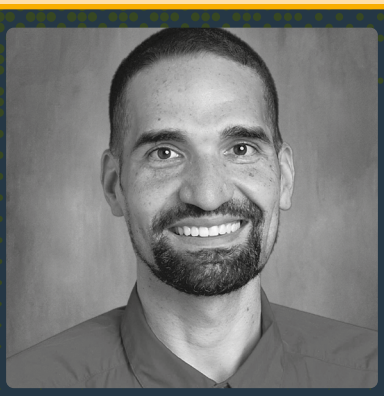
The story I want students to carry

At the end of this project, I did not just want students to walk away with a finished game; I wanted them to walk away with a new understanding of themselves. I wanted them to know that struggle does not disqualify them from creating, that difficulty does not erase brilliance, and that the places where learning feels hardest can also become the places where purpose begins. When students use Canva AI to design games around the skills that challenge them most, they are doing more than practising using technology or creating content; they are reclaiming confidence and building agency. They are learning that they do not have to wait for someone else to create solutions for them. They can be the creators. (HW)

BUILDING A MULTI-AI SYSTEM

How moving from single prompts to a multi-AI workflow helped **Alex Luciano** plan, differentiate, and stay in control of instruction for multilingual learners

I used to ask AI tools for lesson plans. At first, this felt like a breakthrough. I could generate activities quickly, get ideas for differentiation, and save time while planning. But the more I used it, the more I noticed that something was off. The lessons were technically correct, but they weren't mine. The language didn't match my students, the scaffolds didn't reflect what my multilingual learners needed, and the instruction felt disconnected from the reality of my classroom.



ALEX LUCIANO

Alex is a bilingual elementary teacher and adult ESL instructor in Central Islip, New York. His work focuses on integrating AI tools as thinking partners in instruction while maintaining teacher judgement and voice. He shares practical classroom approaches to support multilingual learners and helps educators move from using AI for answers to using it to support deeper thinking (**linkedin.com/in/alex-luciano**).

When the answer stopped working

The problem wasn't that AI was wrong, but that I was using it like a tool that would give me finished answers. I was asking for complete lesson plans and expecting them to fit, when in reality, teaching doesn't work that way. Good instruction comes from knowing your students, adjusting in the moment, and making intentional decisions.

That realisation pushed me to change how I worked with AI. Instead of treating it as one system that should do everything, I began to treat it as a set of roles that could support different parts of my thinking. One interaction helped me generate ideas as a planner, another helped me refine language as an editor, another helped me clarify explanations as an explainer, while yet another helped me check alignment with standards and goals. Nothing came out finished, and every step had a clear purpose.

The toolbox approach

At first, I tried to keep everything in one place by assigning roles within a single conversation. Instead of forcing one tool to do everything, each AI system began to support a different part of my thinking. Like tools in a toolbox, each part serves a specific purpose, and the work moves between them depending on what the moment requires. That shift made the process clearer, more repeatable, and easier to refine without losing my voice. Over time, this settled into a simple rhythm I now think of as diagnose, draft, and calibrate.

Staying in the process

What made the difference wasn't better prompts. It was staying in the process — actively shaping, refining, and adjusting until it felt right for my classroom. The responses began to reflect my patterns more closely, and I needed to change less each time. Even then, the goal was never to step back from the thinking. This is the same shift I started to see in my own planning. AI tools are most useful not when they generate answers, but when they help clarify thinking that is already grounded in the classroom.

This process became most visible in my work with multilingual learners. Instead of accepting generic scaffolds, I could adjust language, simplify explanations, and rebuild examples in ways that made sense for my students. In science lessons, a generated explanation of something like ecosystems or energy transfer might be accurate, but still too dense, so I would reshape it — shortening sentences, replacing abstract terms with familiar language, and building examples that connected to what my students already understood. The final product looked less like something generated and more like something designed.

Guiding vs. receiving

At the same time, I had to become more aware of when to slow down. AI makes it easy to move quickly, but speed can hide weak thinking. Learning to pause, question, and revise became part of the process. Those pauses became my guardrails — small

A multi-AI workflow helped me see more clearly what my students need from me

© twinsterphoto/stock.adobe.com

checks that kept my thinking aligned to my students. I began to think of my role less as using a tool and more as directing a process. The AI was one part of that process, but not the whole of it. There were moments when a response sounded right but didn't hold up under closer review. Staying in that position — where I am the one guiding and not just receiving — became essential to making the work meaningful.

Feedback loops

I began to notice that the quality of the final lesson depended less on what the AI tool produced initially and more on how I responded to it. When I treated the first output as something to evaluate rather than accept, my role as the teacher became more, not less active. I was making decisions at each step — what to keep, what to change, and what to discard — and those decisions were grounded in my knowledge of my students.

Over time, this way of working with AI tools began to change not just how I planned, but how I thought about teaching more broadly. It became less about finding the right answer quickly and more about building the right process. The technology did not make decisions easier; it made my role in those decisions more visible.

That shift also helped me see more clearly what my students need from me. They do not need perfectly generated materials. They need instruction that reflects who they are, how they learn, and where they are in their development. AI tools can support that work, but only when they are guided with intention. When the process is rushed or unexamined, the results may look complete, but they often miss the depth that real learning requires.

In this sense, using AI well is not about efficiency alone; it is about ownership. When that balance is clear, the technology becomes a partner in the work rather than a shortcut around it.

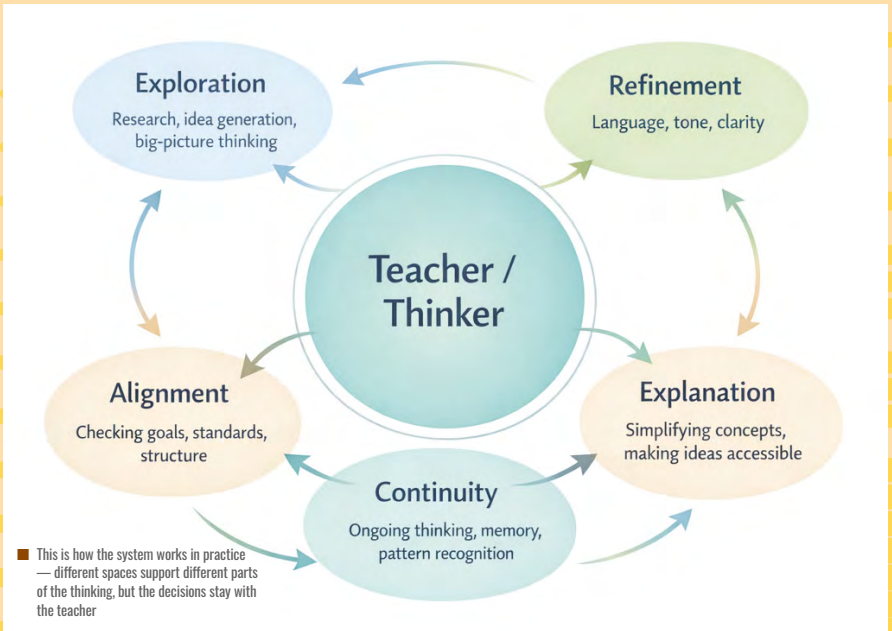
Keeping the thinking with the teacher

For teachers beginning to explore AI, the shift can start in the same way. Instead of asking for a complete lesson and then moving on, pause and treat the first response as a draft. Adjust the language so it sounds like you, reshape the explanations so they fit your students, and check whether the structure supports your objective. That small shift — from accepting to responding — keeps the thinking with the teacher.

From there, the process can grow naturally. You might begin to separate your thinking into stages, focusing on ideas first, then clarity, then alignment. You might revisit the same piece of work more than once, each time improving it slightly. None of this requires a perfect system; it simply requires staying engaged.

In the end, the goal is not to become more efficient at generating lessons, but to become more intentional in how those lessons are shaped. AI can offer a starting point, but it is you, the teacher, who builds the lesson. (HW)

> I BEGAN TO TREAT AI AS A SET OF ROLES THAT SUPPORT DIFFERENT PARTS OF MY THINKING



This is how the system works in practice — different spaces support different parts of the thinking, but the decisions stay with the teacher

DON'T GIVE ME THE ANSWER!

The role of feedback literacy in strategic AI use

Sam is given a homework question that asks her to explain a poem. She reads the question, feels unsure, and almost immediately pastes it into a large language model (LLM), asking, "Can you help me answer this?" The AI-generated response is clear and reassuring. Sam changes a few words and adds it to her work. She has not copied an entire essay, and she may not feel she has violated her school's academic guidelines. But she has not sat with uncertainty, tested her own ideas, made mistakes, or worked out what she does and does not understand. What Sam has done is taken a shortcut in her own thinking process.

Over time, this kind of use can become a habit that researchers call cognitive offloading (**helloworld.cc/singh**, **helloworld.cc/fan**) in which students outsource their problem-solving, critical thinking, and creativity skills to AI tools and LLMs. Within the AI literacy team at the Raspberry Pi Foundation, these concerns have led us to explore what knowledge and skills young people require if they are to engage with AI tools critically without becoming overreliant on them.

Feedback literacy

An important step in building critical-thinking skills in students is teaching them how to interpret information around them, judge whether it is useful, and decide what to do with it, instead of just passively receiving information. This is also referred to as feedback literacy (**rpf.io/fl-qr**) in the research field. For example, if a teacher comments "Develop this paragraph further" on a student's essay, the student needs to decide what that feedback means, whether they understand it, and what action they should take next to improve their work.

In educational contexts, students often receive and interpret feedback in digital, verbal, and written formats. This feedback can support learning in different ways, often through different feedback types:

- **Telling**: when someone is given the answer or correction straight away
- **Guiding**: when someone is given hints, prompts, or steps to help them work towards an answer
- **Challenging**: when someone is asked to question, extend, or justify their thinking

See if you can spot which feedback type this LLM output has generated in **Figure 1**.

You might notice that this particular output has explicitly provided the answer, creating a plan for your introduction instead of helping you think it through. As it happens, LLMs often default to providing the answer, usually in a 'tell' format. As young people increasingly turn to LLMs for schoolwork (**helloworld.cc/pew-ai-teens**), they now have instant access to answers for questions that we would rather have them think through for themselves.

Are you learning if you're given the answer right away?

Being given the answers to questions right away can at times be useful, but students can quickly become overly reliant on this approach, especially if they use LLMs for homework. Deep learning benefits from experiencing different feedback types at different stages of the learning process (**helloworld.cc/hattie-timperley**). For example, students need to be instructed directly when they are learning something new, guided when they are building their confidence and understanding, and challenged when they want to check whether they have understood or mastered something well.

Creating learning-centred prompts

One way educators can support this understanding is by helping students reflect on when each type of feedback is most useful. For example, students could place 'telling', 'guiding', and 'challenging' feedback along a learning timeline. Activities like these can help them recognise that effective learning is not about always receiving the easiest form of support, but about understanding what kind of feedback helps them think and learn most effectively at different moments in the learning process.

Educators can help students stay actively engaged in their learning and be reflective about their LLM use by teaching them how to use LLMs strategically and access different kinds of support when learning (**helloworld.cc/singh**). By tweaking the prompt in an LLM slightly, they can elicit LLM

FURTHER READING

Singh et al. (2025). Protecting Human Cognition in the Age of AI. *Tools For Thought at CHI '25, Yokohama, Japan.* (**helloworld.cc/singh**)

Fan et al. (2024). Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *British Journal of Educational Technology, 56*(2), 489-530. (**helloworld.cc/fan**)

Pew Research Center (2026). How Teens Use and View AI. (**helloworld.cc/pew-ai-teens**)

Hattie & Timperley (2007). The Power of Feedback. *Review of Educational Research, 77*(1), 81-112. (**helloworld.cc/hattie-timperley**)

Oyserman et al. (2006). Possible Selves and Academic Outcomes: How and When Possible Selves Impel Action. *Journal of Personality and Social Psychology, 91*(1), 188–204. (**helloworld.cc/oyserman**)

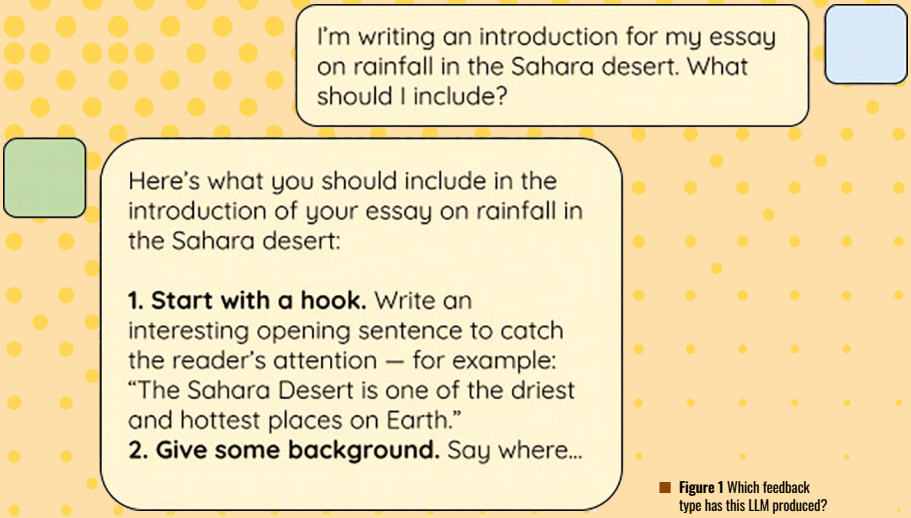

I'm writing an introduction for my essay on rainfall in the Sahara desert. What should I include?
Here's what you should include in the introduction of your essay on rainfall in the Sahara desert:
1. Start with a hook. Write an interesting opening sentence to catch the reader's attention — for example: "The Sahara Desert is one of the driest and hottest places on Earth."
2. Give some background. Say where...

Figure 1 Which feedback type has this LLM produced?

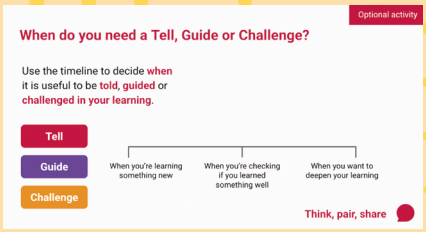
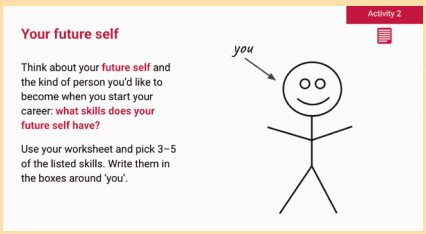

Optional activity
When do you need a Tell, Guide or Challenge?
Use the timeline to decide when it is useful to be told, guided or challenged in your learning.
Tell
Guide
Challenge
When you're learning something new
When you're checking if you learned something well
When you want to deepen your learning
Think, pair, share

Figure 2 The new LLM unit is designed to help young people develop personal principles for responsible AI use


Activity 2
Your future self
Think about your future self and the kind of person you'd like to become when you start your career: what skills does your future self have?
Use your worksheet and pick 3–5 of the listed skills. Write them in the boxes around 'you'.
you

outputs in 'guide' and 'challenge' formats. For example, instead of prompting "I'm writing an introduction for my essay on rainfall in the Sahara desert. What should I include?", students could reframe their prompts to invite guidance and challenge and include key phrases such as "Ask me questions about ...", "Assess my knowledge about ... then ...", or "Help me think in more detail about ...". In this sense, students are using their increased metacognitive knowledge — reflecting on what they know, what they do not know, and what kind of support they need — to write more effective prompts. This can help students engage more deeply with their own learning and turn LLMs into powerful tools for guided enquiry, reflection, feedback, and self-directed learning.

Building future-ready skills

To build strong and lasting critical-thinking and metacognitive skills in students, it is important that students don't just focus on the short-term impacts of LLM use on their learning, but also on the long-term habits they are developing through this use. Activities centred around students' "possible" or "future selves" (**helloworld.cc/oyserman**) can influence students' motivation, goal-setting, and day-to-day learning behaviours by making the consequences of cognitive offloading and uncritical LLM use more concrete and personally meaningful. For example, students could identify the skills they hope to develop in their future careers, such as creativity, problem-solving, or presenting, and then consider whether different ways of using LLMs are strengthening or weakening those future skills.

Activities like these are shown to influence students' motivation, goal-setting, and present-day learning behaviours. They also encourage students to think about the long-term consequences of how they learn and become more critical about their current LLM usage. From there, students can create their own principles for responsible AI use that are helpful to their learning and will help build their future skills, such as fact-checking outputs, avoiding over-reliance on AI-generated answers, and staying actively engaged in their thinking.

New LLM unit

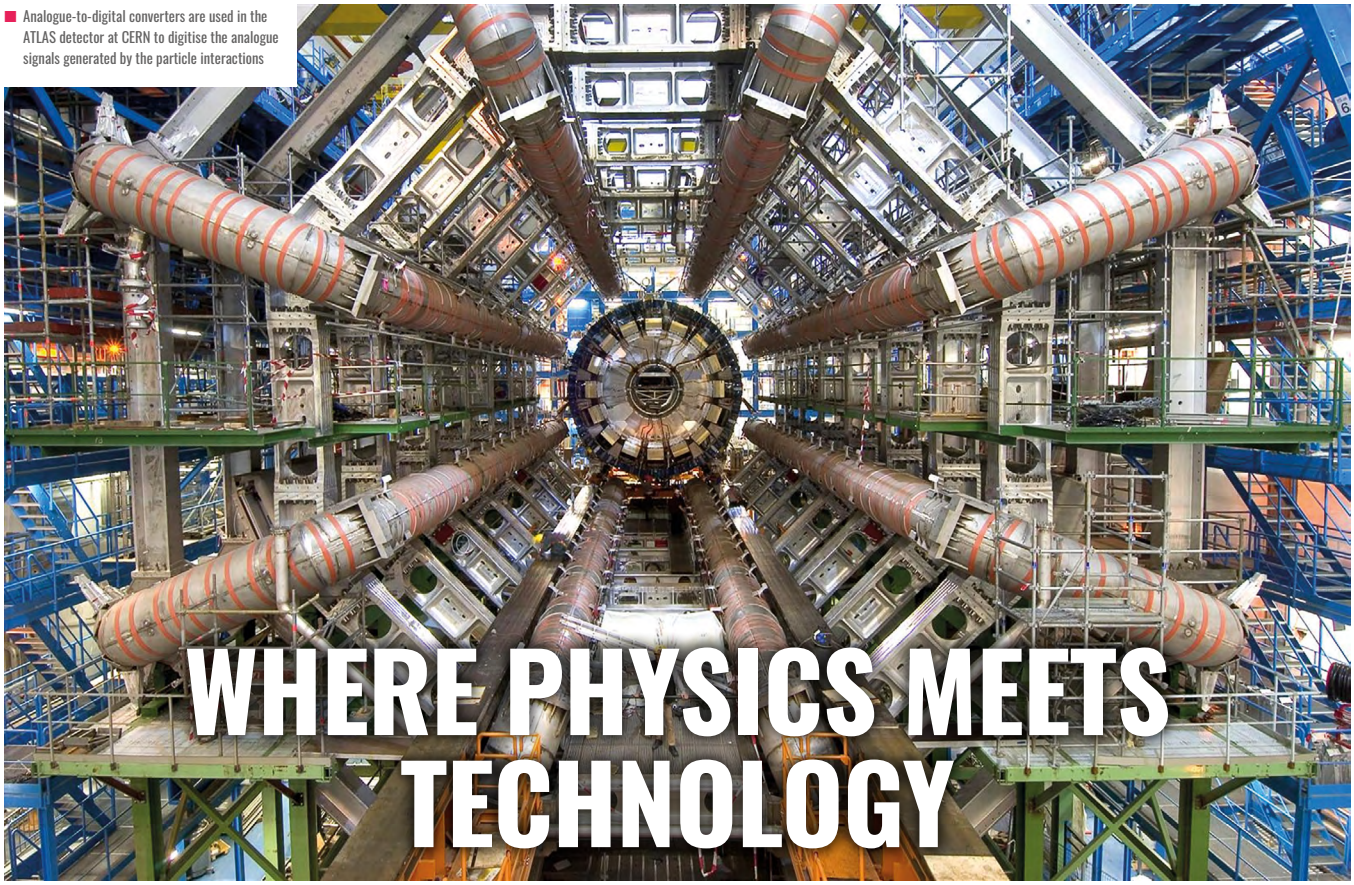
If any of these activities or ideas resonate with you, my team and I at the Raspberry Pi Foundation have developed a new unit on LLMs to build students' capacity to use them strategically in their learning (**Figure 2**). Designed for young people aged 13–16 — or anyone who is of the legal age to use AI tools — the unit aims to help students remain active participants in their own learning in an age of increasingly capable AI systems. Rather than treating LLMs simply as answer-generating tools, it helps students decide whether, when, why, and how to use LLM-generated support, while remaining active authors of their own learning.

The unit draws on the latest research on metacognition, critical thinking, feedback literacy, and learning-centred prompt design. It helps students understand how LLMs work, judge the usefulness of AI-generated output, prompt for guidance and challenge rather than immediate answers, evaluate outputs for accuracy and bias, and develop personal principles for responsible use.

The resources have already been piloted with educators and learners around the world and will be released very soon. For further updates, visit **experience-ai.org/en/units**. We hope our resources will help support a generation of students who can use LLM tools thoughtfully, strategically, and without shortcutting the thinking process that makes learning meaningful. (HW)

REHANA AL-SOLTANE

Rehana is a learning manager at the Raspberry Pi Foundation, creating research-informed AI literacy resources for millions of students worldwide. With a master's from Harvard and MIT, she combines pedagogy, computing, ethics, and critical thinking to help young people understand and shape AI.



Analogue-to-digital converters are used in the ATLAS detector at CERN to digitise the analogue signals generated by the particle interactions

© CERN – copyright.web.cern.ch

WHERE PHYSICS MEETS TECHNOLOGY

Transforming the analogue world into zeros and ones

Taking measurements is a fundamental part of physics work. Although electronic devices are widely used to take measurements nowadays — from the light sensor of a mobile phone to particle detectors — digital measuring instruments are not usually used in schools because of the technological difficulties involved (materials, teacher training, time, and so on).

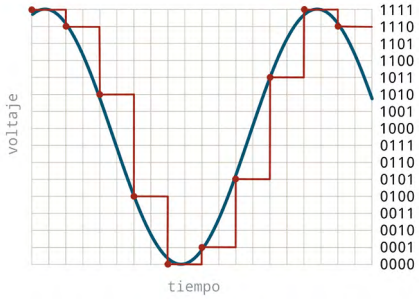
Given the importance we place on a comprehensive and integrated education in the sciences, the physics and computer science departments at our school designed the interdisciplinary activity 'Physics meets Technology: Transforming the analogue world into zeros and ones' for secondary-school students (ages 16–18).

Using Raspberry Pi Pico and the MicroPython programming language, students programmed a temperature sensor in a completely practical project that involved various concepts related to the use of digital measuring devices and the analysis of experimental data. This activity prepares students to work with digital tools and ensures that the contributions of physics and technology can mutually benefit each other.

Hardware and software

MicroPython — Python for microcontrollers — was chosen as the programming language for this activity. We felt that this was the most suitable choice for transitioning from block-based programming to a text-based language, as the students already had experience of programming in Python. As they had never worked with object-oriented programming before, an introductory session was held before the activity to explain basic concepts. Thonny (**thonny.org**), a free, open-source Python IDE that is easy for beginners to use, was used as the development environment.

In terms of hardware, we chose Raspberry Pi Pico 2 W microcontroller and

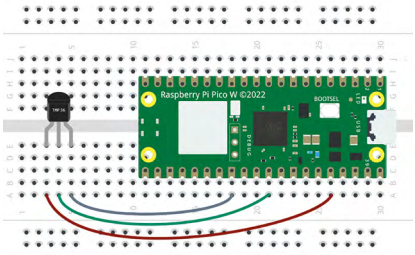


Analogue signal (blue) and its conversion into a four-bit digital signal (red)

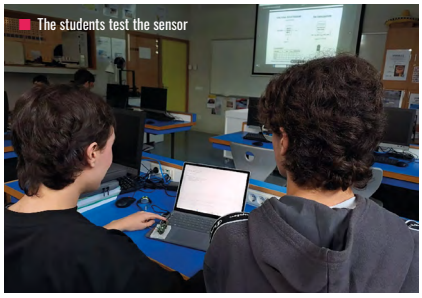
the TMP36 analogue temperature sensor, along with a breadboard and jumper wires to make the electrical connections. A multimeter was also used to demonstrate how temperature relates to the voltage measured by the sensor.

Developing the activity

The activity, which took place over five 50-minute lessons, was divided into three sections. The first section covered general topics relating to sensors. Although the students were already familiar with the



The TMP36 temperature sensor connected to Raspberry Pi Pico



The students test the sensor

term 'sensor', they discovered that the function of sensors is to convert physical signals (in this case, temperature) into electrical signals (voltage). By analysing the sensor's datasheet, they gathered all the technical information required for its use: connections, operating range, the relationship between temperature and voltage, and so on.

The second section focused on addressing a fundamental issue in working with sensors: how to convert a continuous analogue signal, such as voltage, into a digital signal that the microcontroller can understand. The answer to this question is analogue-to-digital conversion, or ADC.

> THE STUDENTS ACQUIRED KNOWLEDGE AND DEVELOPED TRANSFERABLE SKILLS

To explore this concept, the students digitised an analogue signal by hand, which helped them visualise key concepts related to signal digitisation, such as sampling frequency, ADC resolution, and quantisation error.

In the third section, the concepts covered in the previous sessions were put into practice. The circuits were assembled, and the MicroPython concepts required to program the sensor were introduced. Finally, the students wrote a program to measure the temperature automatically. Upon observing fluctuations in the measurements obtained, they sought various strategies to reduce signal noise. One of these involved creating a new program based on the previous one, in which they took several measurements and calculated the average. Now the measurements were practically identical to one another, confirming the reduction in signal noise.

Concepts covered

In addition to the obvious technical content, such as working with hardware and programming in MicroPython, this activity covered a wide range of cross-curricular skills, such as searching for information using reliable sources to find the sensor's datasheet or the board's documentation. Basic mathematical skills were also covered, such as reading graphs, converting from the decimal to the binary number system, linear relationships between quantities, and plotting a line based on its characteristics. Furthermore, within the field of physics, important concepts related to measurement were addressed, such as the difference between direct and indirect measurements, measurement noise, and data smoothing.

Impact and observations

Although this project involved a temperature sensor, the students acquired knowledge and developed transferable skills that they will be able to apply later to other types of projects involving sensors. For example, although they were already familiar with the difference between analogue and digital signals, this activity presented them for the first time with the challenge of converting an analogue signal into a digital one — a skill that is essential when working with analogue sensors. Signal noise and the techniques for minimising it were also new concepts. Learners also discovered the importance of looking up information about a sensor in its datasheet.

The main difficulty we observed was related to the hardware, as learners had problems reading the board's pin map and assembling the circuit. Once these problems were resolved with the teachers' help, the programming part posed no difficulty.

It is worth noting how engaged the students were in the activity and the positive feedback they gave. When they enter the field of physical computing, students come to see programming as a means of interacting with the reality around them, where programmable devices detect signals and act accordingly. This boosts their motivation as they realise that the application has real-world implications. (HW)

ELENA PONCELA & BEATRIZ PADÍN

Elena (left) and Beatriz (right) have been teaching physics, technology, and computer science at high-school level for more than 20 years in Spain and in the USA. They work at Peleteiro School, Spain, and love being involved in international programmes where coding is essential, like Science on Stage Europe (**helloworld.cc/science-on-stage-europe**).

FEATURE

When we ask students to summarise a concept, they have to select, organise, and integrate their knowledge

BEYOND THE CODE: FOSTERING DEEP UNDERSTANDING WITH GENERATIVE LEARNING

Strategies to help students actively select, organise, and integrate knowledge

As educators, we want our learners to recall the factual knowledge of our subject, but more than that, we want them to understand what they have learnt so that they can apply their knowledge to new problems and situations. All too often in the computing classroom, we see our learners completing a task or writing a program but struggling to explain it. This is the problem that generative learning aims to solve.

An overview of generative learning

Generative learning theory was developed by Richard Mayer and Logan Fiorella in the early twenty-tens (**helloworld.cc/generative-learning**). It builds upon cognitive science research and focuses on allowing learners to actively generate meaning and connection from what they are being taught rather than passively learning through copying, memorising, or listening. It focuses on the principle that learning is an active process and as a result, learning is strongest when learners are active, with the knowledge that they are being taught.

Fiorella and Mayer argue that learning happens through three main mental activities: selecting knowledge, organising knowledge, and integrating new knowledge with existing knowledge. These mental models are referred to as the SOI (Select, Organise, Integrate) framework.

- **Selecting knowledge:** similar to abstraction in computational thinking, selecting knowledge focuses on directing attention to relevant knowledge and ignoring irrelevant knowledge.
- **Organising knowledge:** new knowledge is organised into a meaningful structure.
- **Integrating knowledge:** this focuses on integrating new knowledge with existing knowledge. It is arguably the most important part of generative learning and also the hardest to implement effectively in the classroom.

Generative learning in the classroom

Fiorella and Mayer suggest eight generative learning strategies for the classroom:

1. **Summarising** — restating the main ideas of a topic in your own words
2. **Self-explaining** — explaining the content of a topic to yourself
3. **Drawing** — illustrating the content of a topic through images
4. **Teaching** — explaining previously studied topics to others

5. **Self-testing** — testing your knowledge of previously studied content
6. **Mapping** — converting text into a structured arrangement of connected ideas and images
7. **Enacting** — using task-relevant movement to support learning
8. **Imagining** — forming internal images to represent a topic

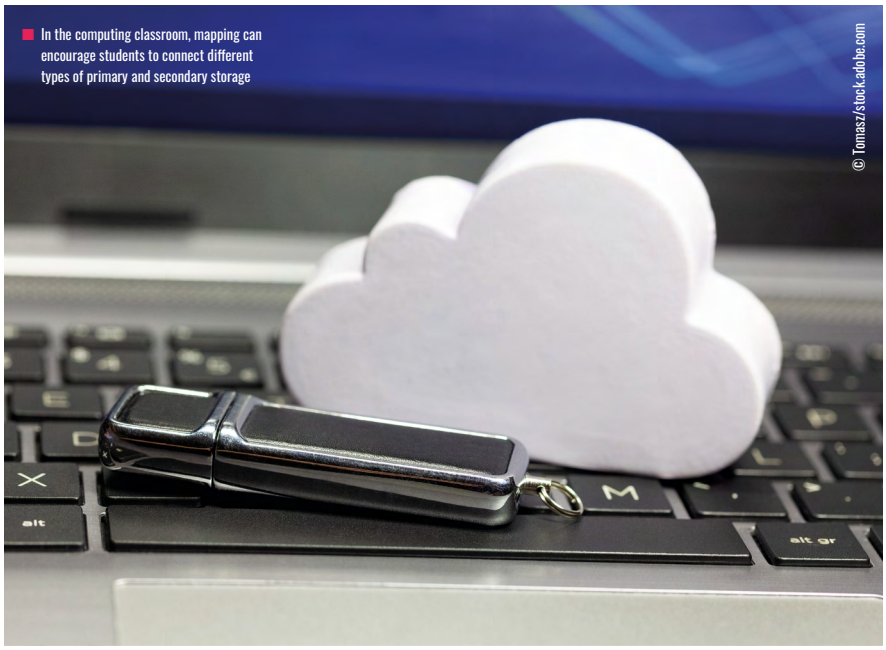
All eight are effective in achieving generative learning in the classroom. I will explore in further detail two that I have found to be effective in computing and which offer an accessible route into implementing generative learning.

Learning by summarising

Summarising is a process that we use all the time, and it is as much part of the human condition as talking itself. When we summarise a film to a friend, we select key points from the film, organise those key points in an order that makes sense, then integrate our knowledge of the film with our knowledge of our friend and explain why or why not we think they might like it.

Summarisation in the classroom is essentially the same process. If you ask a student to summarise a topic, you are asking them to select key knowledge from a topic, organise the selected knowledge into a meaningful structure, and then integrate that knowledge alongside their existing knowledge of the topic.

An example in the computing classroom would be teaching the components of the CPU, assuming that learners already know about the components of a computer. You could ask students to summarise the control unit in 25 words or fewer. By doing this, you are asking them to select key parts of their knowledge on this component, organise them into a meaningful structure, and then integrate that knowledge with their current knowledge of the CPU.



In the computing classroom, mapping can encourage students to connect different types of primary and secondary storage

© Tomasz/stock.adobe.com

Learning by mapping

Learning by mapping is a strategy that enables learners to represent a topic as a woven mesh of intentionally selected knowledge presented as key ideas and images. Relationships between these ideas and images are shown by connecting lines or arrows. Learning by mapping is often oversimplified as 'mind maps', but without careful planning, design, and delivery, these can become passive tasks for learners — the opposite of what we are aiming for.

A carefully constructed mapping task asks learners to intentionally select relevant knowledge, visually organise it, and then make meaningful connections with connecting lines. In the computing classroom, learners could be presented with a piece of text on types of primary and secondary storage. They could then be asked to add these to their maps, grouping and connecting the different types of primary and secondary storage, followed by an additional piece of text on volatility, after which learners would be asked to connect volatile and non-volatile storage types.

Reflections in my own practice

Adopting generative learning has led to a noticeable shift in how students engage with knowledge. Learning is focused on thinking over doing, with a greater emphasis on learners making sense of what they are learning rather than simply completing tasks. No knowledge is taught in isolation and students are directed towards making connections between new knowledge and what they already know.

This has resulted in students becoming more deliberate in the knowledge they select, more structured in how they organise their ideas, and more confident in integrating new knowledge with their existing understanding. I would previously see students completing tasks but struggling to explain the knowledge behind them; I now see students who are able to explain what they have done and why. This is evident both in their verbal responses and in their written work.

It is important to remember that generative learning is only successful when the focus is on meaning, not just completion. Ultimately, it has shifted my classroom away from task completion and towards hard thinking and deep understanding. (HW)

OLIVER BROTHERHOOD

Oliver is head of computing at Penrice Academy and the CELT Trust curriculum lead for computing. He teaches across computer science, media, and enterprise. He is particularly interested in curriculum design, programming education, and how evidence-informed practice can help students build stronger understanding in the computing classroom.

FROM DATA TO DECISIONS

Pongprapan Pongsophon describes how learners can use micro:bit, Python, and machine learning to create predictive classroom systems

Many classroom projects with sensors stop at a simple rule: if the temperature is high, turn on a fan; if the soil is dry, turn on a pump. These projects are useful, but they do not show learners how modern AI systems work. Real systems often do more than react to the present; they use data to predict what is likely to happen next.

I wanted my students to experience this shift. Instead of building devices that only respond to current readings, they built small predictive systems. One system predicted future temperature and switched on a fan before a model factory space became too hot; another predicted future soil moisture and switched on a pump before plants became too dry.

The aim was not to turn middle-school learners into data scientists in one lesson, but one that was simpler and more important: to help them see the full journey from data collection to prediction, and from prediction to action.

From sensing to prediction

The projects used accessible tools: a micro:bit, simple sensors, a relay module, a small fan or pump, Thonny, Google Colab, Python, and scikit-learn. The workflow was the same in both projects.

First, students turned the micro:bit into a data logger. In the cooling project, it recorded temperature and light readings every ten seconds. In the agriculture project, it recorded soil moisture, temperature, and light readings every thirty seconds. Each reading was printed through USB serial as one row of data.

Next, students used Thonny on a laptop to capture the serial stream and save it as a CSV file. This step was important. Learners saw that data does not appear magically inside a machine learning model; it begins as repeated measurements from the world.

Students then uploaded the CSV file to Google Colab. They used pandas (a Python library for analysing data) to create a future target. For the cooling system, they shifted the temperature column 60 rows forwards. Because each row represented 10 seconds, 60 rows meant 10 minutes into the future. For the irrigation system, they shifted the soil moisture column 30 rows forwards. Because each row represented 30 seconds, 30 rows meant 15 minutes into the future.

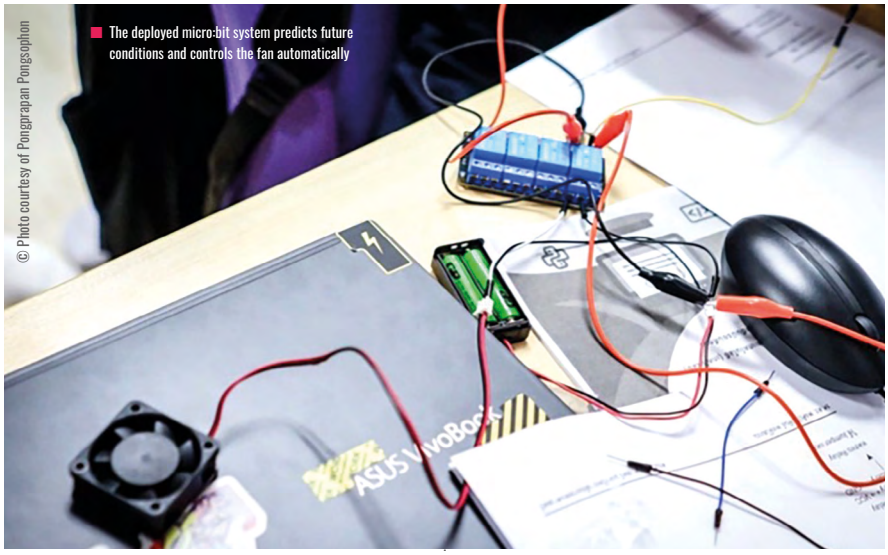
This was the key learning moment. Prediction became visible. Students could point to one row and say, 'These are the current inputs. This is the future value we want the model to learn.'

Training the model

After creating the future target, students trained a linear regression model. I chose linear regression because it is transparent. Students can inspect the learnt weights and the intercept. They can see that the model is not magic; it is an equation learnt from data.

For the cooling system, the model used current temperature and light to predict future temperature. For the irrigation system, it used current moisture, temperature, and light to predict future moisture.

Students then evaluated the model using two statistical measures. Mean absolute error helped them ask, 'How far off are our predictions on average?' R^2 helped them ask, 'How much of the pattern did the model

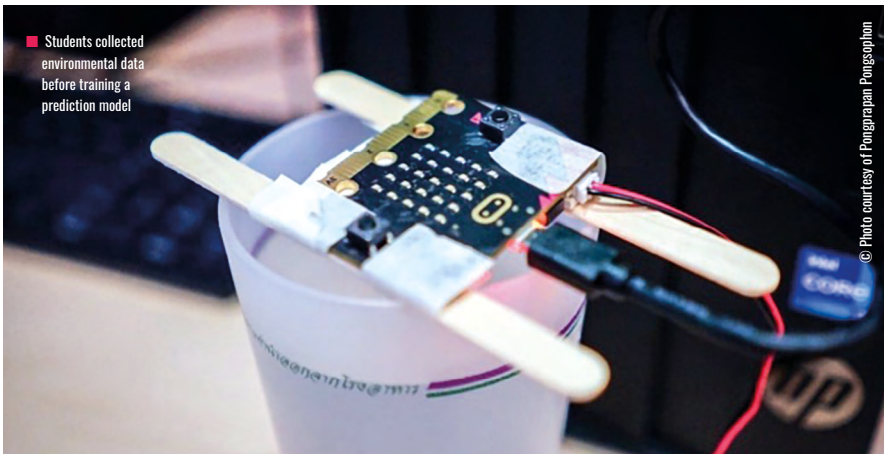


© Photo courtesy of Pongprapan Pongsophon

The deployed micro:bit system predicts future conditions and controls the fan automatically

START SIMPLE

Before connecting a relay, let students focus on prediction itself. Ask them to collect sensor data, create a future target in Google Colab, and display 'HIGH' or 'LOW' on the micro:bit based on predicted conditions. Once students understand how prediction works, add the fan or pump as a final extension activity.



Students collected environmental data before training a prediction model

© Photo courtesy of Pongprapan Pongsophon

> THESE PROJECTS MAKE AI TOOLS MORE UNDERSTANDABLE AND LESS MYSTERIOUS

explain?' These numbers were not treated as abstract statistics — we connected them to system behaviour. A model with a high mean absolute error might switch the fan too late, or turn the pump on when it is not needed.

The most satisfying part came next. Students copied the learnt coefficients and the intercept from Colab into the micro:bit deployment code. The model trained on the laptop now ran on the microcontroller. The micro:bit read live sensor values, calculated a predicted future value, and used that prediction to control the relay.

In the cooling project, the fan turned on when predicted future temperature reached the chosen threshold. In the agriculture project, the pump turned on when predicted future moisture dropped below the threshold. The students were no longer building a reactive device, but a system that used present data to take early action.

Classroom insights

This activity changes the role of coding in the classroom. Coding is not only a way to give instructions to a machine; it becomes a way to model a relationship between the present and the future.

It also changes the role of sensors. Sensors are not only input devices for immediate control, but become tools for building data sets. Students learn that the quality of the model depends on the quality, range, and relevance of the data they collect.

The projects also give teachers a practical way to teach AI without treating it as a black box. Students collect the data, train the model, inspect the equation, deploy it, and test the system themselves. This makes AI tools more understandable and less mysterious.

There were some misconceptions. Some students at first thought prediction meant certainty. Others thought that the model would explain the cause of changes in temperature or moisture. These were useful teaching moments. We discussed the difference between prediction and explanation, and why models can fail when the environment changes.

The activity works best with strong scaffolding. Learners need step-by-step support for serial data capture, CSV files, row shifting, model training, and deployment. I found it helpful to keep the first model simple, use a short data set, and focus discussion on the system pipeline rather than on advanced machine learning theory.

I usually run the activity in five stages:

- 20 to 30 minutes of data logging
- 15 to 20 minutes of preparing the CSV file
- 15 to 20 minutes of training and evaluating the model
- 15 to 20 minutes of deployment and testing
- 20 minutes of discussion and reflection

Teachers can adapt the same workflow to many contexts. Students might predict classroom noise, light levels, plant growth conditions, air quality, or energy use. The important pattern remains the same: collect data, define a future target, train a model, deploy it, and evaluate how prediction informs action.

For me, the value of the project does not lie in the fan or the pump, but in the thinking behind them. Learners see that computing can connect data, models, and decisions. They learn that AI is not only something they just consume, but something they can build, test, question, and improve. (HW)

FULL RESOURCES

Teachers who want the full technical appendix, wiring guide, Python scripts, Google Colab workflow, and deployment code can access the resources by scanning this QR code.

PONGPRAPAN PONGSOPHON

Pongprapan is an associate professor of science education at Kasetsart University, Thailand. He teaches computing-integrated STEM, data science, and machine learning for education, with a focus on practical classroom projects using Python, sensors, and microcontrollers.

© kaveee29/stock.adobe.com

CAN AI TOOLS HELP TO MARK A-LEVEL COURSEWORK?

Surprising insights from the use of AI systems in assessment

Like many computer science teachers, I have a complicated relationship with computer science non-exam assessments. On one hand, it is arguably the most authentic part of the A-level computer science course (for students aged 17–18). Students get to build something that matters to them, solve real problems, and demonstrate the programming and development skills they have spent two years learning. However, they tend to produce large reports, and I find myself working through source code, screenshots, testing evidence, design documentation, and evaluation sections. The mark scheme is detailed, but applying it consistently across an entire cohort requires a significant amount of time and concentration. There is also the dreaded wait over the summer to see if the marking was upheld by the exam board and your judgements were accurate and within tolerance.

Like every teacher, I want my marking to be fair, consistent, and evidence-based. But assessment inevitably involves professional judgement. Two experienced teachers can look at the same piece of work and spend a considerable amount of time discussing whether a particular criterion has been fully met, and when you are within one or two marks that is acceptable, but what if the marks are out of tolerance? This can be due to interpretations of the mark scheme, bias, or even the past experiences of the teacher.

That got me thinking about whether AI tools could be used to help us assess, and whether they could do this in a meaningful and accurate way. The point here is not to replace teachers or make final decisions, and certainly not to automate assessment. But could AI systems act as second markers? Over the years, I have known departments of non-specialists and even specialists that do not have access to other teachers with the expertise to offer support. Could AI tools be the answer?

The question

Much of the discussion around AI in education focuses on how students use the system. Will they use it to cheat? Should we ban it? How do we detect AI-generated work? Those are important questions, but I found myself asking different ones: What happens when teachers use AI tools? Can a carefully designed AI system review coursework, apply assessment criteria, and provide meaningful feedback to support

professional judgements? The only way to find out was to build one and test it.

Building the system

Rather than purchasing a commercial platform, I decided to create my own workflow. The system combined OpenClaw (**openclaw.ai**), a large language model linked into DeepSeek AI and utilising Telegram as an interface. The aim was not simply to generate a mark, but to analyse evidence, apply assessment criteria, and justify every decision made.

Students' reports, source code, testing evidence, screenshots, and supporting documentation could be submitted to the system. The AI tool would then review the evidence against the relevant assessment criteria and produce a structured assessment outcome. I was sceptical and frankly thought I might be wasting my time.

What interested me most was not the mark itself, but whether the tool could explain why a mark had been awarded and identify evidence supporting each judgement. Could it produce reasoning that would make sense to an experienced assessor? Those questions became the focus of the project. Installing and configuring the system was straightforward, and using DeepSeek significantly reduced the costs involved.

Teaching the tool what good assessment looks like

One of the biggest misconceptions about AI is that you can simply provide a mark scheme and expect meaningful results. That is not how it works, and assessment is more nuanced than that. Experienced teachers develop an understanding of standards through years of marking, moderation discussions, and exposure to previous student work. We get better every time we mark, and the same is true for AI.

I wanted the AI tool to benefit from the experience I have had over the years and make sure it had a basis for making judgements, not just a rigid framework in the form of a mark grid.

To achieve this, I used approximately four years of historical assessment material consisting of around 60 previously assessed projects. These included examples across a range of standards alongside the associated marking decisions and teacher commentary.

All material was anonymised before it was used. No student names, candidate numbers, or personally identifiable information were included. The intention was not to teach the AI system specific answers, but to show it how the decisions were made, and to expose the system to examples of how assessment criteria are interpreted in practice.

In many ways, it was like supporting a new teacher through their first moderation cycle. You do not simply hand them a mark scheme and hope for the best. You discuss examples, review previous work, and talk through assessment decisions, sometimes line by line. My hope was that the AI tool would benefit from a similar process.

> "COULD A CAREFULLY DESIGNED AI SYSTEM REVIEW COURSEWORK AND GIVE FEEDBACK?

A NOTE ON GDPR

Whenever AI is discussed in schools, data protection concerns rightly follow close behind. From the outset, I established a simple rule: no student names, personal identifiers, candidate numbers, or other information that could identify an individual would be submitted to the system.

Projects were anonymised before processing. The AI tool assessed the work itself, rather than the student who produced it.

This was important not only from a General Data Protection Regulation (GDPR) perspective, but also because it removed the potential for unconscious bias.

Any school considering similar approaches should carefully review its own policies, data protection requirements, and safeguarding obligations.

AI may offer exciting opportunities, but those opportunities must always be balanced against privacy, security, and professional responsibility.

My first surprise

I expected the system to struggle, and in fact I was almost hoping it would. Assessment is often presented as one of the last areas in which human judgement remains irreplaceable. It is a teacher's bread and butter. As teachers, we understand context. We understand quality. We recognise strengths and weaknesses that are not always easy to define.

Experience with marking helps teachers master academic standards; the same is true for AI

I anticipated wildly inconsistent results; instead, something unexpected happened. The AI tool's observations were frequently similar to my own. It identified strengths I had highlighted. It identified weaknesses I had identified. In some cases, it even pointed out missing evidence that I had initially overlooked and later agreed was significant. That was the point at which the project became genuinely interesting. There were some disagreements and areas where the tool misinterpreted the marking criteria and had to be realigned. Once these issues were identified and corrected, the question was no longer whether the tool could apply the mark scheme, but whether it could do so reliably every time.

The moderator test

Teacher agreement is one thing — moderator agreement is quite another.

The process I followed created multiple opportunities for comparison. First, I completed my own assessment using the assessment criteria and completing the feedback sheets, annotating the work, and making my own judgement. The same project was then reviewed by the AI system. After that, the work was independently moderated by a colleague from another school before the results were compared and finally submitted through the normal examination board moderation process. This meant I was not simply comparing the AI tool's judgements to my own judgements — I could compare its conclusions to multiple independent reviewers. One example still stands out. The AI tool assessed one student's project at 70 out of 70, full marks. My immediate reaction was scepticism. Most experienced teachers are naturally cautious about awarding maximum marks. There is always a temptation to look one more time for a weakness, a missing piece of evidence, or a reason to be slightly more conservative. The AI tool, however, was unequivocal. It identified evidence against every criterion and produced a detailed justification explaining why each mark had been awarded.

Rather than accepting the result immediately, I reviewed the project again. I challenged the AI system's conclusions, checked the evidence, and revisited the assessment criteria.

After doing so, I concluded that the evidence supported the award. I may have been inclined to award 68 out of 70, but the justifications the AI tool gave made it very difficult to argue with. The project was then independently moderated by a colleague from the other school, who reached the same conclusion. Finally, the work progressed through the normal examination board moderation process and the 70/70 mark was upheld.

That experience changed the way I viewed the system. The AI tool was not simply generating numbers; it was forcing me to challenge my own assumptions as an assessor. In this instance, the AI was prepared to award full marks based solely on the evidence presented, whereas my instinct was initially to be more conservative. The final outcome suggested that its confidence had been justified.

> "I ANTICIPATED WILDLY INCONSISTENT RESULTS; INSTEAD, SOMETHING UNEXPECTED HAPPENED

Humans and AI working together can make assessment more consistent and reliable for everyone involved

What the AI tool did well

Several strengths became apparent throughout the project.

Consistency

Teachers are human. By the 20th project, most of us are more tired than we were during the first, our feedback becomes less detailed, and our output decreases. The AI tool did not suffer from that problem. Every project received the same level of scrutiny, regardless of when it was assessed.

Evidence tracking

The system was particularly effective at identifying where evidence existed and where evidence was missing. Many students complete substantial work but fail to document it effectively. The AI tool was good at highlighting situations where evidence was implied rather than demonstrated.

Acting as a second marker

This became the most valuable aspect of the project. I rarely accepted the AI tool's conclusions without question. Instead, I treated them in the same way I would treat feedback from a trusted colleague. If it agreed with me, it provided reassurance. If it disagreed, it encouraged me to review the evidence again. That process improved my confidence in the final decisions.

What the AI tool did not do well

The project also highlighted some important limitations. The AI system was strongest when it was applying explicit criteria, but it was less effective when dealing with qualities that experienced teachers often recognise instinctively. Innovation, elegance, originality, and creative problem-solving can be difficult to capture within a rigid assessment framework.

Human assessors naturally understand context. We know the student. We know the challenges they have faced. We understand the journey behind the final product. AI does not possess that wider understanding, and for that reason alone, I do not believe AI should replace teachers in assessment, nor do I believe it should make final decisions independently. The teacher remains essential.

What this means for education

My experience suggests that the most promising role for AI in assessment is not automatic marking. Although I had success at getting the system to generate completed feedback forms and automatically highlight assessment grids, the most promising role for AI in assessment is quality assurance.

Imagine being able to submit a project and receive:

- A provisional assessment
- Evidence linked to individual criteria
- Identification of missing documentation
- Areas requiring further review before final marks are submitted

That would not remove professional judgement; it would strengthen it.

For newer teachers or non-specialists, it could provide reassurance. For experienced teachers, it could provide a valuable second opinion. For departments, it could improve consistency.

Lessons learnt

Looking back, the most important lesson was not about technology, but about expectations.

I began the project expecting to discover the limitations of AI. Instead, I discovered that AI was capable of far more than I had anticipated when provided with clear criteria, structured evidence, historical examples, and carefully designed prompts. At the same time, the experience reinforced my belief that assessment is fundamentally a human activity. The best results of all, though, came when AI and teacher expertise worked together. Neither was as effective on its own.

> “ ASSESSMENT IS OFTEN PRESENTED AS ONE OF THE LAST AREAS IN WHICH HUMAN JUDGEMENT REMAINS IRREPLACEABLE

Final thoughts

The arrival of AI in education has generated understandable concern. Many teachers are worried about what it means for assessment, coursework, and academic integrity. I understand those concerns, and I share some of them.

At the same time, though, I believe we have an opportunity. Used carelessly, AI can create problems. Used thoughtfully, it can become one of the most powerful tools available to educators. My experiment did not convince me that teachers should be replaced; it convinced me that teachers should be supported.

Any use of AI in assessment must be accompanied by appropriate safeguards, including anonymisation, GDPR compliance, transparency, and ongoing human oversight.

The future of assessment is unlikely to be ‘humans versus AI’. It is far more likely to be humans and AI working together to make assessment more consistent, more efficient, and ultimately more reliable for everyone involved.

While I was writing this article, Pope Leo XIV published his first encyclical, *Magnifica Humanitas*, addressing the impact of artificial intelligence on society (**helloworld.cc/humanitas-work**). Although this is written from a very different perspective, one of its central themes resonated strongly with my own experience: AI should serve humanity, not replace it. The document repeatedly stresses the importance of preserving human dignity, human responsibility, and meaningful human judgement in a world increasingly influenced by artificial intelligence.

My experience with AI-assisted assessment has led me to a similar conclusion. The most effective approach was never to hand responsibility to the machine. The greatest value came when AI acted as a knowledgeable assistant, helping me to review evidence, challenge assumptions, and make more informed decisions as a teacher.

So, what does the future of assessment look like? It is likely to be a case of humans and AI working hand in hand, with teachers always remaining responsible for the decisions that matter most. (HW)

AARON CONNOR

Aaron is head of computer science at Lancing College and has over 15 years’ teaching experience. Winner of the 2015 West Sussex Inspirational Teacher of the Year award, he also holds an MEd and is currently undertaking a PhD. He runs the Teacher of Computing YouTube channel supporting students and educators and is a passionate advocate of lifelong learning, drawing on his own experience of dyslexia.

FEATURE

© Bohdan/stock.adobe.com

FROM THREE LIVES TO INFINITE LOOPS

We can show our young people how to code systems that reward skill rather than just time spent

What the evolution of play means for young coders

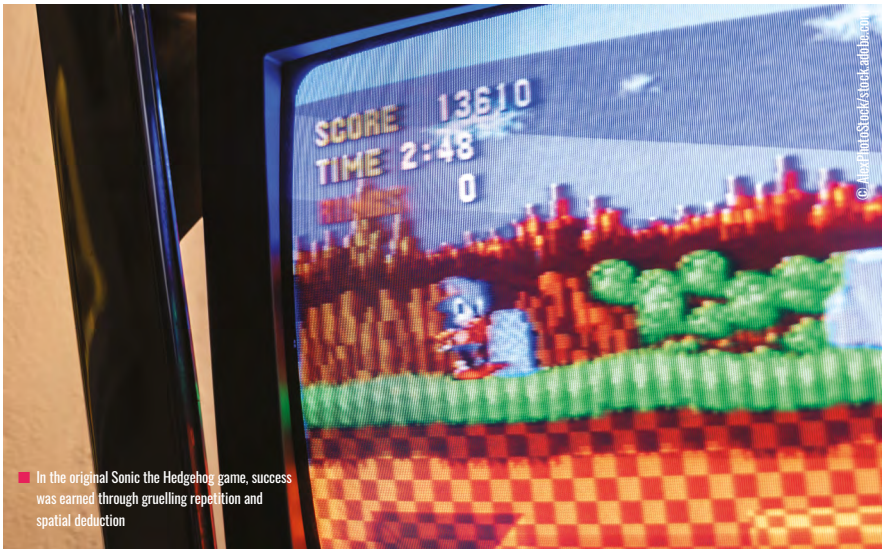
If you grew up gaming in the late 1980s or 1990s, you likely carry a specific kind of muscle memory — the frantic, rhythmic tapping of a button as your last life flickered on a CRT screen. Whether you were navigating the unforgiving physics of Super Mario Bros., chasing rings in Sonic the Hedgehog, or exploring the sprawling hubs of Spyro the Dragon, the digital contract was clear: you typically had three lives, no auto-save, and no glowing GPS arrow pointing towards the exit. Success wasn't guaranteed by participation — it was earned through gruelling repetition and spatial deduction. You didn't just play a level; you memorised its heartbeat. Today, our Code Club students walk into a vastly different landscape — one defined by seamless flow, cloud saves, and quality of life features that, while making gaming more inclusive, have fundamentally altered the cognitive load required to succeed.

The art of persistence vs. frictionless flow

Classic platformers were masterclasses in frustration tolerance. Because a single mistake could send you back to the very beginning of a world, you learnt to analyse enemy patterns and time your jumps with mathematical precision. This cycle built a grit that is often missing in modern titles in which auto-saves trigger every 30 seconds. When we mentor young creators in our clubs, we have a unique opportunity to reintroduce meaningful challenge. Instead of allowing students to code a text-heavy tutorial that explains every mechanic, we should encourage them to design discovery levels — stages where the player is forced to learn a new ability, like a double-jump or a shield, through environmental cues and safe failure. By doing so, we teach our students that a game's value isn't just in the winning, but in the persistence required to get there.

Internal maps vs. digital guidance

Think back to the mental effort required to navigate the original Legend of Zelda or the labyrinthine corridors of DOOM. Without a floating waypoint or a mini-map peppered with markers, players were forced to build high-fidelity mental maps. Neuroscience suggests that this type of spatial navigation actively strengthens the hippocampus — the brain's engine for memory and learning. Modern wayfinding, such as detective vision or glowing paths on the floor, reduces friction but also bypasses the need for independent problem-solving. In our sessions, we can steer students away from these digital crutches by challenging them to build exploration-based projects that rely on landmarks or visual storytelling. When a student codes a secret passage marked only by a subtle change in wall texture, they aren't just making a game; they are exercising their player's spatial reasoning and rewarding their curiosity.



© AlexPhotoStock/stock.adobe.com

In the original Sonic the Hedgehog game, success was earned through gruelling repetition and spatial deduction

> A GAME'S VALUE ISN'T JUST IN THE WINNING, BUT IN THE PERSISTENCE NEEDED TO GET THERE

The end state vs. the endless loop

One of the most profound shifts in gaming is the disappearance of the Game Over screen. In the 1990s, games had a natural end point; when your lives were gone, the console often went off. Today's ecosystems, such as Fortnite or Roblox, are designed as infinite loops with no inherent stopping points. While this keeps engagement high, it makes self-regulation difficult for a developing brain. We can address this head-on by discussing ethical design with our young coders. We should guide them to build natural breaks into their code — intentional pause points such as a shop interface, a lore-heavy dialogue scene, or a clear Mission Accomplished screen that provides a logical moment to step away. By intentionally designing for a satisfied player rather than a hooked one, our students learn that great software respects the user's time.

Mastery vs. constant feedback

Modern gaming is often built around rapid dopamine delivery: loot boxes, XP pop-ups, and daily login bonuses create an intense feedback loop. This isn't inherently bad, but it can overshadow the deeper reward of delayed mastery — the feeling of finally conquering a difficult boss after a week of practice. To counter this, we can show our young people how to code systems that reward skill rather than just time spent. Instead of a daily free gift for logging in, we might help them script a Mastery Medal that unlocks only when a player completes a level without taking damage. This shifts the focus from 'What can the game give me?' to 'What can I achieve within the game?'

Designing with intent

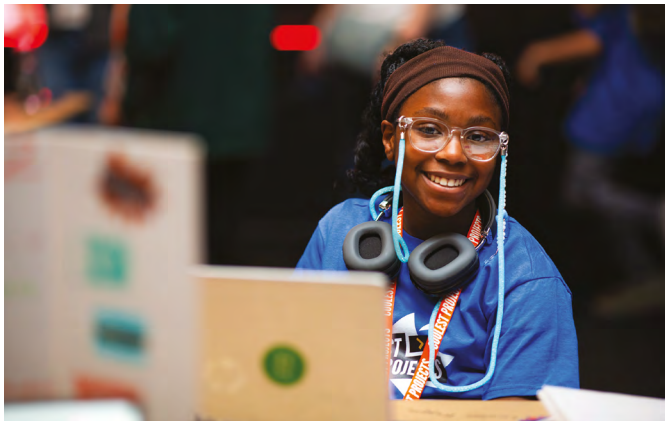
The evolution of games offers a powerful case study in how design influences human behaviour. At Code Club Australia, we aren't interested in nostalgia for its own sake, but in empowering children to be the architects of the future. By helping them understand how mechanics have changed — and the psychology behind those changes — we move them from being passive consumers to intentional creators. Whether they are coding three lives or an infinite loop, we want them to realise that their code carries values. When they choose to design for challenge, for mental mapping, and for well-being, they aren't just levelling up their coding skills; they are learning to build a digital world that adds value to people's lives. That is the real level-up. (HW)

© Tomasz/stock.adobe.com

Infinite loops in games like Roblox, with no inherent stopping points, keep engagement high but make self-regulation difficult

KAYE NORTH

Kaye is a passionate STEM educator based in Queensland, Australia. She currently serves as the community and engagement manager for Code Club Australia. She has taught in both primary and high schools and has spent many years working with school leaders to map out their whole-school STEM journeys. Kaye actively works to build teacher capability and to support schools in building professional learning communities to establish a network of colleagues on similar STEM journeys (**@KayeNorth1**, **linkedin.com/in/kaye-north**).



COOLEST PROJECTS UK AND IRELAND

Celebrating the judges' favourite digital creations made by young people at our annual showcase

Coolest Projects is a worldwide celebration of the amazing things that young people make with code. There's an annual online showcase where over 15,000 young creators share their projects, and in-person events in 16 countries too. Over 300 young people attended events in Dublin and Bradford in spring 2026, and judges picked a few of their favourite projects to highlight at each event. The UK and Ireland's 2026 favourites are an incredible selection!

Using technology, solving problems

Many young creators used technology to solve problems that really matter to them. Arittra, 16, from Bunclody, created Cara, an AI companion for immigrants in Ireland that helps users access health services and understand legal rights. Arittra created his project to help immigrants like his parents. "One of the key problems with AI is that they hallucinate, but I found a countermeasure to it. I trained it on search variety and gave it back-up data, so if one part of the server starts hallucinating, another part automatically steps into its place to find maximum accuracy." Maksym, 10, in the UK, also wanted to help new arrivals to the country feel at home with his project 'My New Home: Adventures Around Sheffield'. "This Scratch game is based on real places I explored after moving from Ukraine to the UK. Players travel across a map, complete mini-games and quests, and discover how exploring new places can help a new country feel like home."

Sisters Winter-Rose, 11, Summer-Nova, 10, and Autumn-Grace, 7, from Code Club Kilkenny in Ireland had the idea of using technology to help their family too, who were worried about dangerous sun exposure. They created 'Sun Safe Skin', a hardware project featuring a handheld UV sensor with colour-coded LEDs to recommend what UV skin protection should be worn. Chinemerem, 13, from Warrington in the UK also wanted to promote health through coding. He made a Roblox game called 'Immune System Defense', in which players defend a heart from dangerous viruses. He said, "I really enjoyed the event ... One of my favourite parts was seeing people scan my QR code and actually play my game live during the showcase."

'YouTensile', a hardware project made by Ewan, 15, from Bolton, assists people with geriatric tremors, Parkinson's disease, and other involuntary tremors to eat independently with ease. Ewan used gyroscope module values to counteract tremors rapidly with precision.

Seth from Wales showcased his 'PopBot Reminder' system, inspired by his nan's need to set reminders for his granddad, which can now be activated by face recognition. "Our facial recognition system kept getting confused, so we took more photos to help it recognise better. We used different backgrounds, clothes, and lighting to help it recognise our faces better."

Collins, 11, from Darlington, and Inshirah, 15, from Epsom, both focused on helping others study. Collins' web calculator helps solve complex maths problems, and event attendees couldn't find a problem too hard for it all day! Inshirah created an AI-powered chatbot that helps students with their studies by providing clear, accurate, and age-appropriate responses. She focused on using a friendly and engaging tone to make learning easier, more accessible, and more enjoyable for younger learners.

High-powered hardware

Aine, 11, and Michael, 14, from Longford Code Club in Ireland, enjoyed getting creative with hardware when making their project 'PiForge', and they also wanted to help others to use hardware. PiForge is an AI-powered block coding platform for Raspberry Pi. It compiles to C for performance, runs on a load-balanced cluster, virtualises GPIO access, and supports extensions, allowing scalable, collaborative hardware programming.

Jay, 15, from Eldon Code Club, attended Coolest Projects UK for the third year in a row, having participated in the online showcase since 2021. He created 'MindsPi', a project about electromagnetic brainwaves and the control of robots using the mind. Lucas, 13, from Bradford, also had fun experimenting with hardware with his project 'The sm@rt antique PI-cture!', an electronic reconversion of an early 1900s camera using Raspberry Pi. He included a specially designed flash and built-in Raspberry Pi camera while taking special care not to damage the original vintage camera.

Just for fun

Creators like Felicia, 11, and Francesca, 8, from Glasgow, had fun making their projects, which were also designed to bring joy to others. The sisters made a Scratch game called 'Accidental Adventure', in which the player is trapped in a glitchy 3D maze. They incorporated a micro:bit to create a real industrial remote. Bobby, 10, from Tramore Code Club, created 'Bobby's Irish Counties Quiz' to test knowledge while having fun. "This is a Scratch game that tests how well you know your Irish geography. It shows an outline of each county, which you have to name, and it times how long it takes you to name them all. It also constructs the map of Ireland as you go."

Niharika, 9, from Belfast, made a project inspired by her hobby. "I love Irish dancing, so I made 'Irish Dance Studio' in Scratch! Pick a jig or type of reel and watch my dancer perform with matching music and costume. I drew all the costumes and coded all the buttons myself!" Gabriel, 12, from Bacup in the UK, also showcased a game inspired by his passion. 'Fossil Roulette' showed the process of fossilisation as a game of chance and encouraged players to try and change from a living creature to a fossil. Gabriel also shared some of his real fossil collection on event day.

Creators like Naomi, 10, from London, made games and animations that were popular with other coders. Her game 'Life of Scratch' follows a humorous day in the life of Scratch Cat, with surprises, cool effects, and diamonds to collect. Emer, 12, from Gorey Code Club in Ireland, created a Scratch game called 'Let's have some fun' that allows you to choose different fashionable outfits for your character. Leo, 11, from Dublin created an incredible platform game called 'Pixels Perilous Parkour'. He blew the judges and fellow creators away with his incredible innovations in Scratch. The game 'Attack of the Demons!' was made by Stephen, 12, from Gorey Code Club. "This is inspired by the game 'Doom'. You are going around dungeons fighting demons, collecting loot and weapons, and finding keys to open doors and chests."

From hardware to games and from solving problems to having fun, the 2026 UK and Ireland creators made amazing projects that blew us away. If you'd like to see and play more projects from young people worldwide, head to the 2026 online showcase gallery (**helloworld.cc/cp-2026**) and prepare to be impressed. (HW)

> "YOUNG CREATORS USED TECHNOLOGY TO SOLVE PROBLEMS THAT MATTERED TO THEM

HELEN GARDNER
Helen is the programme manager for Coolest Projects at the Raspberry Pi Foundation.

FEATURE

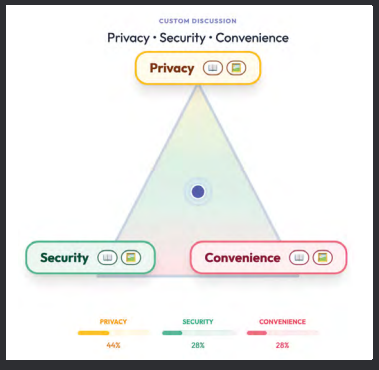
Students planning to make a video

MANAGING COGNITIVE LOAD FOR DEEPER THINKING

Using structured scaffolding, discussion, and decision-making to help students organise ideas, explore complex questions, and communicate clearly

TRIANGULAR THINKING HACK

Although essential questions are the gold standard when it comes to planning, there is a quick hack to generate these kinds of discussions, and that is to use a priority triangle. You can pick any three words and ask which is the most important. See examples at **helloworld.cc/priority-triangle**.



When we manage cognitive load, we aim for a state of flow in which the task is not so difficult it overwhelms students, but not so easy that it is boring and offers no engagement. A good starting point is to think of essential questions for your topic. Questions such as 'Should facial recognition be used in schools?' or 'Who is responsible when an automated system gets something wrong?' invite students to weigh ideas rather than simply recall facts. They also bring a challenge for teachers: if the thinking is too open too early, students can become overloaded before the discussion has properly begun.

Essential questions

A powerful essential question is carefully framed. It should be open enough to allow multiple defensible positions, but focused enough to guide comparison. It should centre on a genuine tension, for example safety versus privacy, convenience versus fairness, or efficiency versus bias. Strong questions are specific to a context, use clear language, and make it possible to identify and weigh distinct factors. Most importantly, they lead naturally into structured thinking, where ideas can be organised, compared, and revised rather than answered immediately. More examples of effective essential questions include:

- To what extent can surveillance in the name of safety be justified?
- When an algorithm knows exactly what you like, are you choosing your interests or is the app choosing them for you?
- If life becomes 100 percent frictionless because of tech, do we lose the skills that make us smart?
- Can a computer ever be truly fair if it is learning from a world that is already full of bias?

JAMES ABELA

James is director of digital learning and entrepreneurship at Garden International School in Kuala Lumpur, Malaysia. He is founder of the South East Asian Computer Science Teachers Association and **ReadySetCompute.com**.

- When technology transforms the world, who profits and who pays the price?
- If a self-driving car or a medical chatbot makes a mistake, who is to blame?

These questions invite comparison, trade-offs, and justification rather than just searching for a single correct answer.

Too much at once

Essential questions should be challenging, and they might not have any real answers, but they are a starting point for discussion.

A number of tools are available that will help scaffold thinking and encourage discussion. These include a diamond 9 (and the simpler diamond 5) activity, debate-style elimination activities, and line rankings. A set of these for computing can be found at **helloworld.cc/thinkrank-studio**. Each of these can be done think–pair–share style or in small groups to encourage thinking.

Imagine a lesson on computer network components. Rather than beginning with a full-class explanation of routers, switches, servers, and protocols, students could start with five key components in a diamond 5 or line ranking, for example: routers, switches, servers, firewalls, and access points. Students could spend three minutes ranking these based on importance in keeping a network running before discussing their reasoning with a partner. This reduces the number of variables and gives them a structure for comparison before any deeper explanation begins.

Students debating the value of social networks

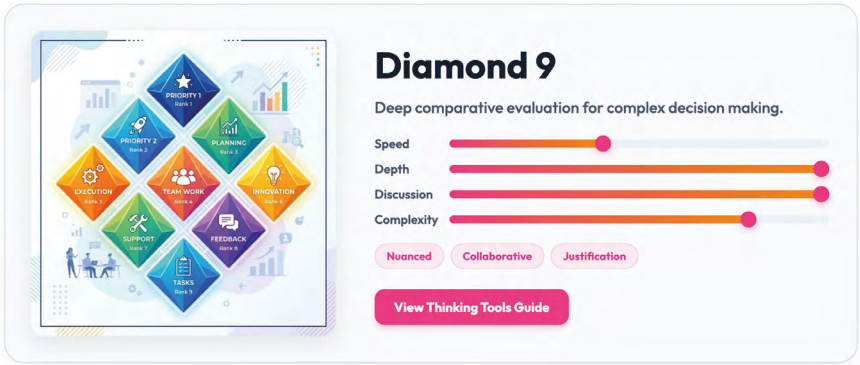
Wrestling with ideas

After ideas have been organised, students can begin to apply their understanding to more complex questions. Open-ended or ethical issues are especially useful here because they require judgement rather than just recall.

Students might move from ranking five statements about facial recognition to answering a broader question such as whether facial recognition should be used in schools. With scaffolding in place and based on the quality of discussion, teachers can choose to increase the challenge further. Students might expand their categories into advantages, disadvantages, stakeholders, and long-term consequences.

They could then use that thinking as the basis for a class debate, a piece of extended writing, a presentation, or a more creative outcome such as an infographic or short video. At this stage, students may begin to produce something, but the value lies in the fact that the production grows out of structured comparison, discussion, and revision rather than immediate reaction.

By intentionally managing cognitive load through scaffolding, we do more than just make a lesson ‘easier’. We provide students with the mental architecture required to navigate an increasingly complex world. When we move from the overwhelm of a blank page to the structured rigour of a diamond 9 or an essential question, we aren’t just teaching computer science — we are teaching discernment. In an era in which answers are automated and information is infinite, the ability to organise, weigh, and justify ideas is the most human skill our students can attain. (HW)



Students organise and rank ideas, making their thinking visible through structured discussion

LESSON PLAN

AI AROUND YOU

The following lesson plan is taken from 'Module 2: Solving problems with AI', a five-lesson unit from Experience AI. It is aimed at learners aged 11 to 14 and builds on the learning from 'Module 1: Understanding AI', to help young people begin to solve real-world problems using AI tools.

ABOUT EXPERIENCE AI

Experience AI is an educational programme that offers cutting-edge resources on artificial intelligence and machine learning for teachers and school-aged learners. Developed in collaboration between the Raspberry Pi Foundation and Google DeepMind, the programme supports teachers in the exciting and fast-moving area of AI, and gets young people passionate about the subject. Sign up for a free account at **experience-ai.org**.

Experience AI

AGE RANGE

11–14 years

LEARNING OBJECTIVES

- Understand the differences between predictive and generative AI
- Describe the types of problems AI can help solve
- Recognise the ethical issues of developing AI applications

Explore how AI is used in everyday life

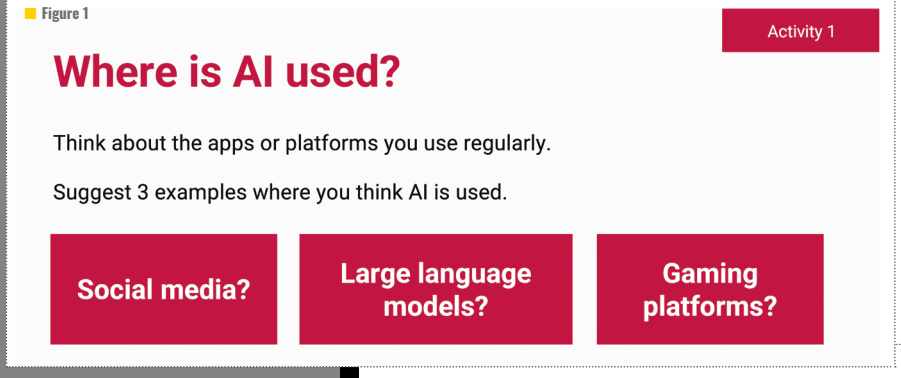
In this lesson, taken from the Raspberry Pi Foundation's Experience AI resources (**experience-ai.org**), students will explore how AI is used in everyday life and learn to tell predictive and generative AI apart. They will investigate problems they can solve using AI and reflect on when AI tools are an appropriate way to solve a problem.

ACTIVITY 1: WHERE IS AI USED?

5 MINUTES

Prompt learners to come up with examples of AI in everyday platforms they use. If they find it hard to think of examples, use the options on the slide to help them (**Figure 1**). To prepare for the next section, guide them to sort their examples based on whether the AI tool predicts outcomes or creates content.

Figure 1


Activity 1
Where is AI used?
Think about the apps or platforms you use regularly.
Suggest 3 examples where you think AI is used.
Social media?
Large language models?
Gaming platforms?

REQUIREMENTS

- Resources from Module 2, Lesson 1 of the Experience AI resources (**helloworld.cc/ai-around-you**):
  - Slides
  - Lesson plan
- Worksheet 2: solving problems with predictive AI
- Worksheet 3: ethical and data considerations in AI systems
- Worksheet (optional): ethical inspector

ACTIVITY 2: SOLVING PROBLEMS WITH PREDICTIVE AI

20 MINUTES

You will guide learners in discussing the differences between generative and predictive AI. You do not need technical knowledge of how the applications work.

Introduce the terms 'generative AI' and 'predictive AI' by reading the definitions on the slide (**Figure 2**). Emphasise that generative AI generates new content, while predictive AI analyses past data to predict future outcomes.

Ask learners to categorise the examples they came up with in Activity 1 to ensure that they understand the difference between the categories.

Ask learners what types of output a generative AI tool can produce (**Figure 3**). Briefly explain that large language models (LLMs) are a type of generative AI that generates written content based on the user's prompt. Explain that generating images is another application of generative AI.

Use the examples in **Figure 4** to highlight the three different areas where predictive AI could be used: entertainment services, science and medicine, and emergency forecasting. The slides in this lesson contain more detailed examples (**helloworld.cc/ai-around-you**).

Figure 2

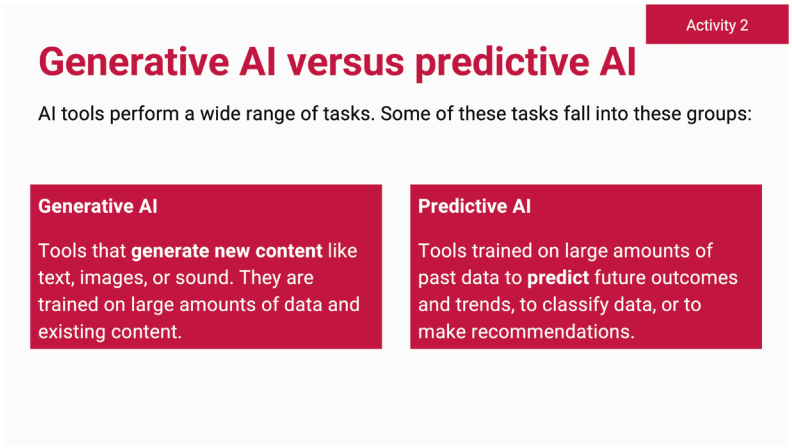

Activity 2
Generative AI versus predictive AI
AI tools perform a wide range of tasks. Some of these tasks fall into these groups:
Generative AI
Tools that generate new content like text, images, or sound. They are trained on large amounts of data and existing content.
Predictive AI
Tools trained on large amounts of past data to predict future outcomes and trends, to classify data, or to make recommendations.

Figure 3

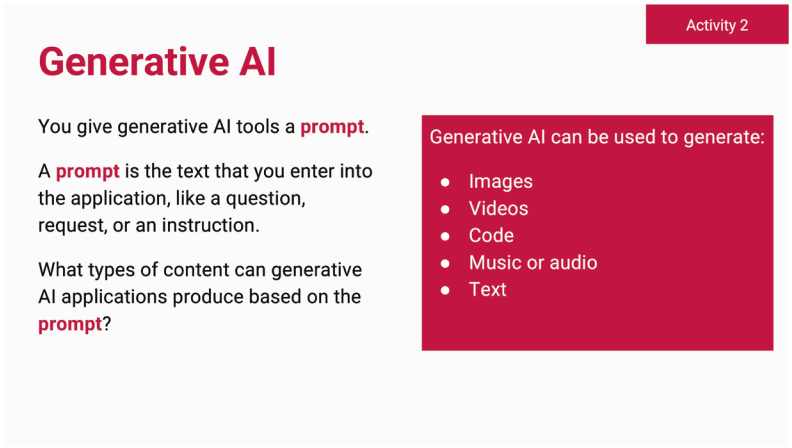
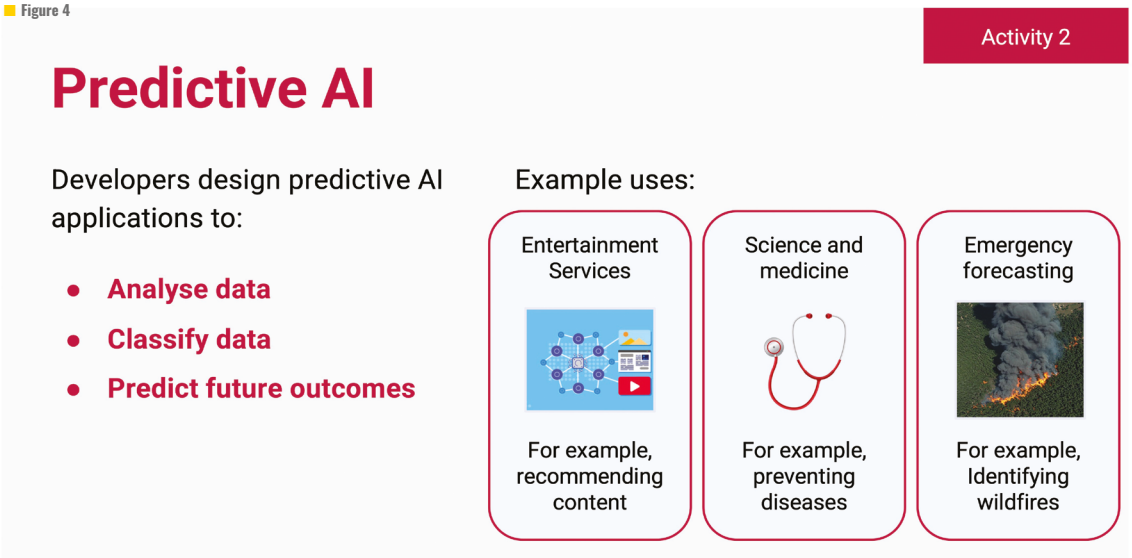

Activity 2
Generative AI
You give generative AI tools a prompt.
A prompt is the text that you enter into the application, like a question, request, or an instruction.
What types of content can generative AI applications produce based on the prompt?
Generative AI can be used to generate:
Images
Videos
Code
Music or audio
Text

Figure 4



ACTIVITY 3: ETHICAL AND DATA ISSUES

15 MINUTES

Introduce the activity and explain that learners will imagine themselves as developers of a data-driven AI system, who have to consider important questions (**Figure 5**). Divide your class into groups of three to five learners. Hand out one copy of **Worksheet 3** to each group (**helloworld.cc/ai-around-you**). Explain that each group will choose one out of three example AI tools on their worksheet to consider ethical and data concerns about the tool. You can also use the slides to show the three example AI tools: AI chatbot for mental health support, personalised learning AI tutor, or AI content moderation for social media.

Read out the questions on **Figure 6** and use the **think, pair, share** pedagogical approach with your learners. Depending on how much time you have, you can facilitate a longer discussion.

Figure 5

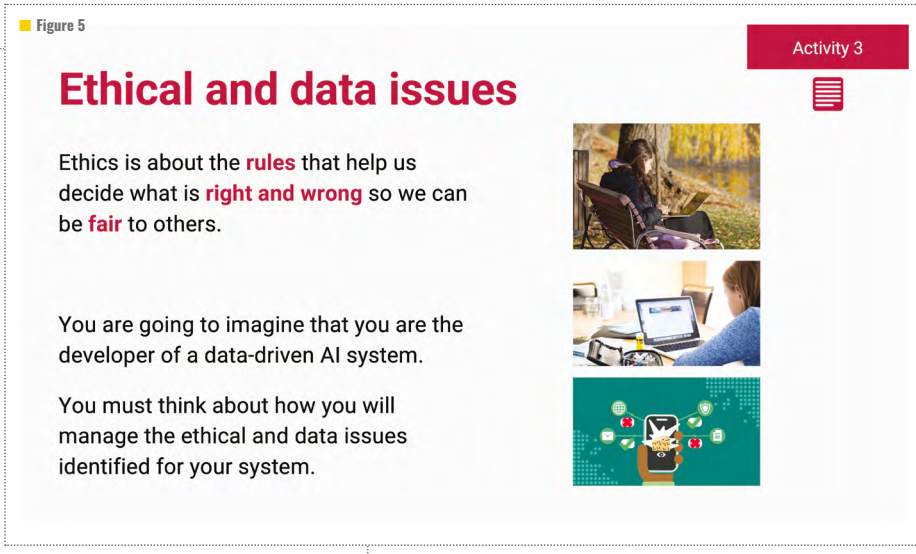
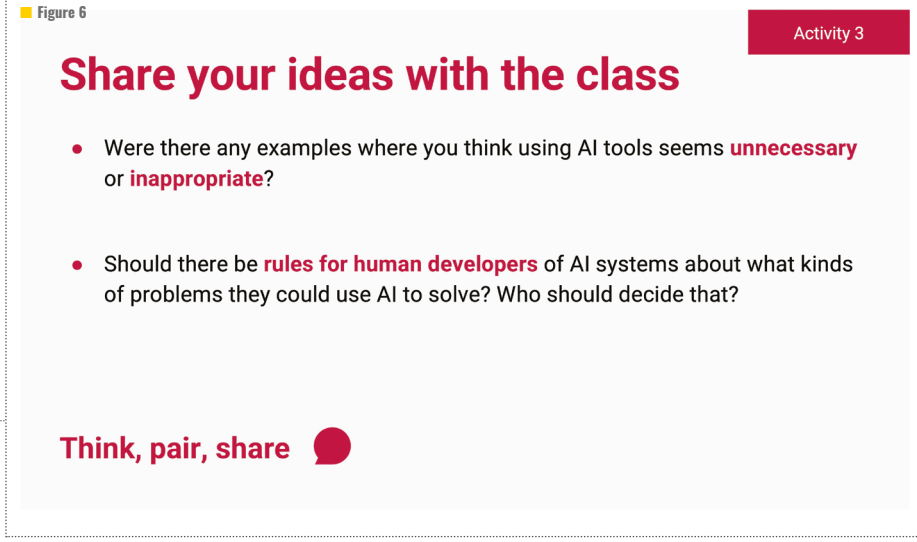

Activity 3
Ethical and data issues
Ethics is about the rules that help us decide what is right and wrong so we can be fair to others.
You are going to imagine that you are the developer of a data-driven AI system.
You must think about how you will manage the ethical and data issues identified for your system.

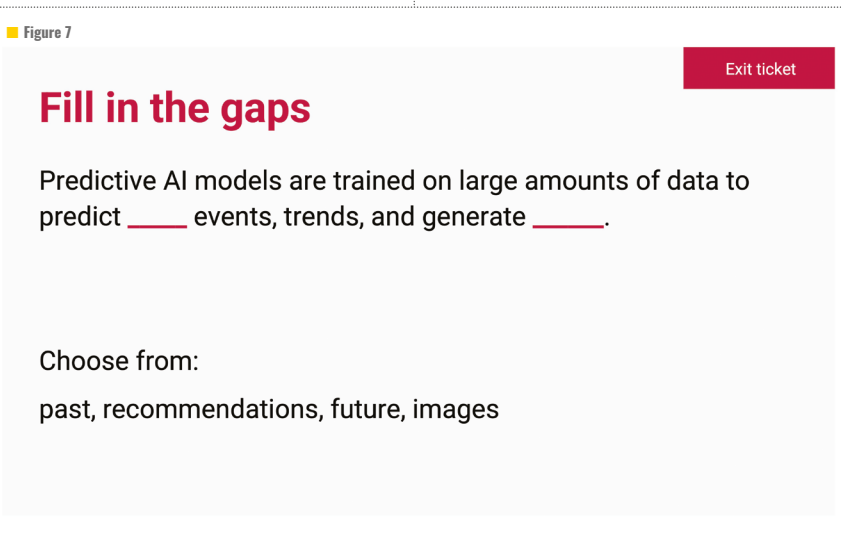
Figure 6


Activity 3
Share your ideas with the class
Were there any examples where you think using AI tools seems unnecessary or inappropriate?
Should there be rules for human developers of AI systems about what kinds of problems they could use AI to solve? Who should decide that?
Think, pair, share

RELEVANT LINKS

experience-ai.org

Figure 7


Exit ticket
Fill in the gaps
Predictive AI models are trained on large amounts of data to predict ____ events, trends, and generate _____.
Choose from:
past, recommendations, future, images

PLENARY: EXIT TICKET AND WRAP UP

5 MINUTES

Figure 7 provides an opportunity for you to assess your learners' understanding of the concepts covered in this lesson. Additional question slides are provided in the full lesson, and are followed by an identical slide with the answer.

The murals at Franklin D. Roosevelt High School in Brooklyn, USA, tell you what each department believes in

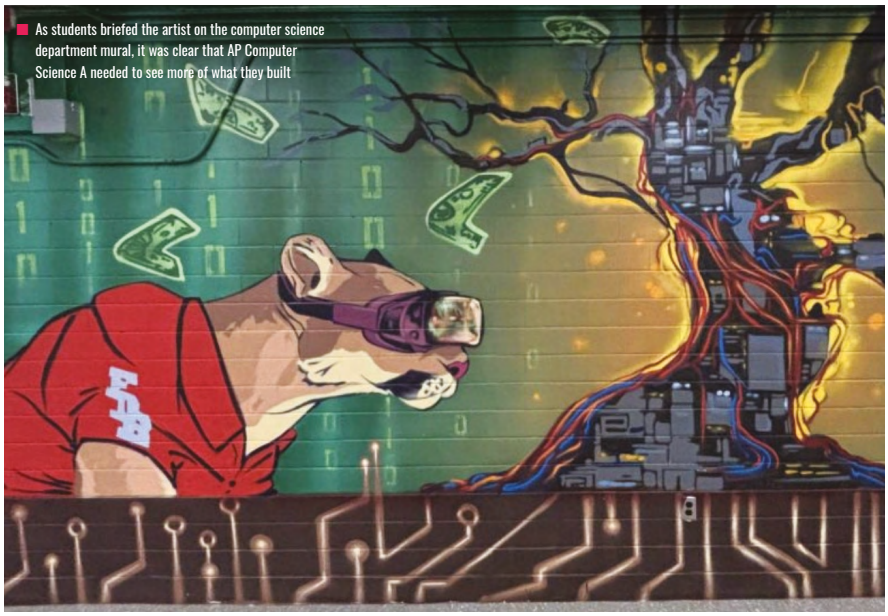
REIMAGINING AP COMPUTER SCIENCE A

Rebranding the AP course to 'Applied Processing for Creative Scripting and Algorithms'

Before you even reach a classroom at Franklin D. Roosevelt (FDR) High School in Brooklyn, the hallways tell you what each department believes in. Each department has a mural designed in consultation with a panel of students. The science department's mural bursts with periodic table elements and DNA helixes. The English department's is layered with literary quotes and open books. The world language mural traces the migration of the monarch butterfly, a nod to the journeys so many FDR students and their families have made themselves. And then there is ours: two cougars (FDR's mascot), one wearing a VR headset, eyes wide inside a digital world, and another rendered as Tony Stark, surrounded by drones and robots and the quiet suggestion that the future is something you can build. Just before you reach my classroom, the hallway is already making an argument about what computer science can be.

The mural came out of a conversation. A panel of students met with the artist during the design process, and the artist asked my students a simple question: What connections does programming have to their lives? Students mostly circled back to one thing: a project near the end of the year when they had coded Conway's Game of Life in Processing (**processing.org**), an open-source, Java-based language and environment designed to help beginners create visual and interactive programs easily. That one project where we coded a grid of cells flickering and evolving according to simple rules was the one project that stuck. I took that personally, not as a failure, but as a signal that my students needed a more immediate connection to computer science.

At FDR, and I suspect at most high schools, AP Computer Science A is not a course for students who have already decided they are going to be software engineers. Some will go on to study computer science, while others are here for a credit on a transcript. My job is not just to prepare them for an exam, but to make the subject feel like something that is worth caring about. If the only project that genuinely landed, after a full year of instruction, was the one where they could see what they had built, the message was clear. Processing belonged at the beginning of the year, not the end.



As students briefed the artist on the computer science department mural, it was clear that AP Computer Science A needed to see more of what they built

Contextual learning

The shift to Processing did not mean stepping away from the AP curriculum. Every major concept the exam demands found its way into a project, but the entry point was always something students wanted to make. The clearest example was the 2024 AP Computer Science A exam, which included a free-response question about a bird feeder — tracking how many birds visited, managing probabilities, and updating states. A bird feeder does not mean much to a student who grew up in urban Brooklyn. So we kept the computer science and changed the context to a Whack-a-Mole game. The 95 percent chance of a common bird visiting became a 95 percent chance of a target worth clicking; the 5 percent chance of a bear visiting became a penalty to dodge. Students tracked scores, lives, and time remaining, watching those variables drive everything on screen. The concept was identical to the exam question, but the investment was completely different.

A similar energy carried into the recursion unit, which came much later in the year but belongs in the same conversation about engagement. Students first traced factorial calculations by hand, but it was the fractal tree that lit the eyes up. Students modified existing code to customise their tree for Earth Day, changing branching angles and length ratios and adding blossoms or autumn leaves. Watching a tree grow denser or sparser as you adjust a single parameter made the idea of a method calling itself feel not just logical, but alive.

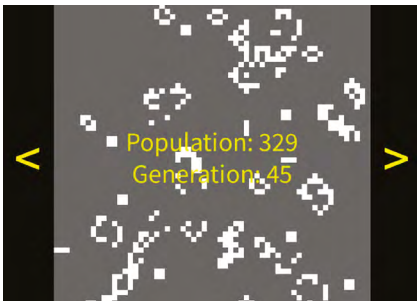
> “USING PROCESSING IS NOT A DETOUR FROM RIGOROUS COMPUTER SCIENCE: IT IS THE ON-RAMP TO A MORE ENGAGING EXPERIENCE

Coding as community building

A familiar project is the robot project in which students work in pairs, with one student giving verbal instructions on how to draw a robot while the other draws exactly what they are instructed to draw. This demonstrates that precise communication is necessary when you are telling a machine what to do. From there, students moved to grid paper, labelling axes and translating their design into coordinates before writing a line of code. When they finally opened Processing, functions were not an abstraction to memorise, but the solution to a problem students had already felt. Structured peer-review cycles put students in front of each other’s work, and curiosity did the rest. Robots were blinking, waving, moving across the screen with key presses that nobody had asked for, and suddenly everyone wanted to know how.

Visualising complex concepts

Two other projects changed how quickly students grasped concepts I had spent significant time reteaching in previous years. The ‘New Year’s Eve confetti simulation’ introduced object-oriented programming (OOP). Each particle had a position, a colour, a velocity, and a behaviour. The alternative of managing five hundred particles without a class was obviously unworkable the moment students tried to picture it. The case for OOP made itself. A candy store project did something similar for ArrayLists. Students crowdsourced candy designs and built a store interface where users could click to add or remove items from a cart. The classic ArrayList bug, skipping an element when removing left to right, had never been



Conway’s Game of Life is a cellular automaton devised by the British mathematician John Horton Conway in 1970

diagnosed so quickly. In a console program, that bug is invisible until you look for it. In the candy store, it showed up immediately: a candy that refused to leave the cart, a visual glitch that made the problem undeniable. Students found the bug faster and understood why it happened, because they could see it.

> MY STUDENTS NEEDED A MORE IMMEDIATE CONNECTION TO CS AND A PROJECT WHERE THEY COULD SEE WHAT THEY HAD BUILT

The Game of Life

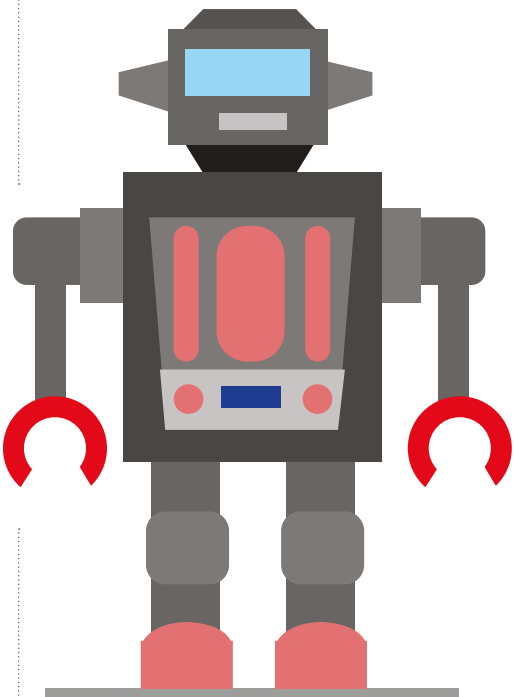
Of course, we still did Conway's Game of Life. It was the project that started all of this, and this year, it was the one I was most curious to see. In previous years, teaching it meant teaching two things at once. Students were meeting Processing for the first time, while trying to hold the object-oriented design of the 'Cell' class, the neighbour-counting logic, the rules for birth and survival, and the challenge of updating the grid without letting one generation bleed into the next. The CS concepts were real and demanding, but they were clouded. This year, Processing was invisible by the time we got there. It was a fluent language rather than a foreign one. What remained was the computer science, and students gave it their full attention. Some were able to go further by customising the rules, building pause and reset controls, or adding colour gradients tracking how long a cell had been alive.

I want to end with the moment that surprised me most, and it has to do with artificial intelligence (AI). Most conversations about AI in CS classrooms centre on the same anxiety: students will use it to write their code, and we will not be able to tell. That anxiety is real, but it assumes text-based output. The canvas was partly a pedagogical choice and partly an honest one: AI struggles to draw what a student imagines, which means the work stays theirs. Around the December holidays, we ran a design challenge: students coded their own gingerbread cookies in Processing, making creative decisions about shape, decoration, colour, and personality. Then we handed the same prompt to an AI tool. The disappointment was palpable. What came back was technically present and completely generic. Sitting right next to their own work, the difference needed no explanation. The creative specificity students brought was not something a language model could generate from a text description. That might be the most honest argument for creative coding in a CS course: it makes student work genuinely theirs in a way that is increasingly rare and increasingly important.

An on-ramp to rigorous CS

I'd like to say this to every teacher running a creative coding elective in Processing, p5.js, or a similar environment: What you are doing is not a detour from rigorous computer science; it is the on-ramp. Students who have spent a semester making things on a canvas already understand objects, state, iteration, and algorithmic thinking in a deeply felt way. That foundation is transferrable. What stands between a creative coding course and AP-level work is not a wall but a bridge, and a shorter one than it might appear. Add the data structures the exam demands. Give students practice with AP-style questions. The creative instincts, the habits of iteration, of debugging by looking, are already there. A well-taught creative coding elective is not the course students take before they are ready for computer science. It is the course that makes them ready for anything. (HW)

Students worked through unplugged problem-solving tasks before coding their robots in Processing



JOSHUA HANS

Joshua is a computer science and mathematics teacher at Franklin D. Roosevelt High School in New York. He is a Math for America fellow, and was secretary of CSTA NYC. This year, he and his students reimagined AP Computer Science A from the ground up, rebranding the course as Applied Processing for Creative Scripting and Algorithms.

FLATGAMES: MAKING GAMES INSPIRED BY ZINES

Becca Rose explores how a corner of indie game culture built on handmade, intimate stories can unlock unexpected and personal making

A few years ago, while on holiday in the Canary Islands, I got a bee stuck in my hair. It buzzed around for a little while, then popped out and stung me in the middle of my forehead.

A few weeks later, I was at my laptop, surrounded by crafting debris, making a flatgame about the bee's journey and the surreal feeling of it stinging me right between my eyes. The game is small and intimate, with hand-drawn inky assets, quickly assembled in 2D in Unity. It's called Apropolis, and you can play it here: **helloworld.cc/apropolis**.

I made it at a flatgame jam with a group of other digital crafters, many of them gamers who wanted to create something personal and playful but felt stifled by the scale and polish of big games.

What is a flatgame?

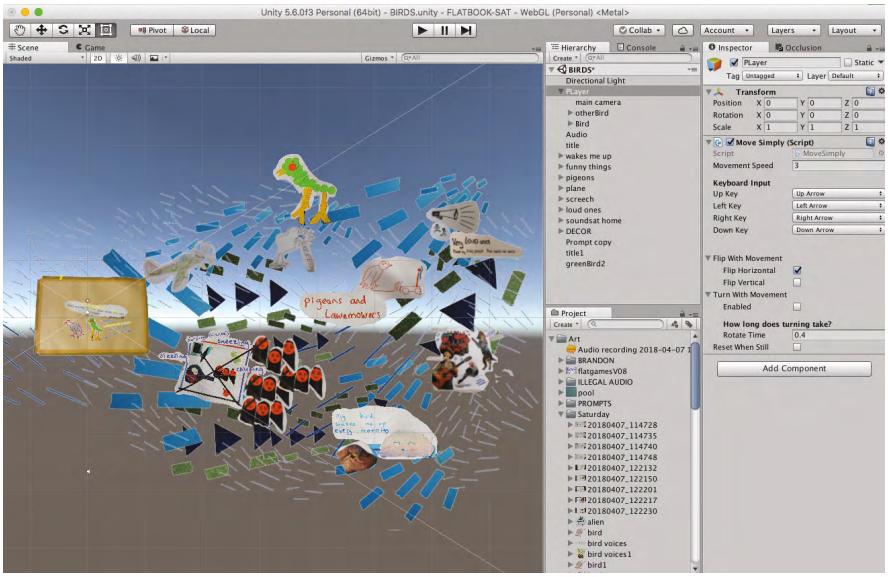
A flatgame could be described as a digital zine or an art game, characterised by personal subject matter, hand-drawn or collaged visuals, and deliberate simplicity. There are no points, enemies, or winning mechanisms. You simply move through them and get to explore a small, personal thing that someone has shared.

The genre has been steered by indie gamemakers such as Dreamfeel (**dreamfeel.ie**) as a small act of rebellion against the idea that games need to be complex, polished, or impressive to count as games. Instead, makers repurpose game engines such as Unity to make small, handmade, and immediate stories. These are part of a wider experimental game culture that showcases artistic games, such as the label A MAZE (**a-maze.net**).

There have been a number of flatgame jams where people come together to make games — some online and hosted through indie game streaming site **itch.io**, others in person at a media centre or community arts space. I've run jams with teenagers using Unity, with 7- to 11-year-olds in Scratch, and with families at games festivals. I've also made a few flatgames myself. They are always about simple, ordinary things — getting stung by a bee, learning to tie knots, or sitting around a campfire.

For young people and creators, the format encourages a creative flurry in a short space of time. The focus on expression rather than gameplay, the limited timeframe, and the crafty aesthetic create a platform for a huge range of games to be made. Young

A flatgame is characterised by hand-drawn visuals and deliberate simplicity



Flatgames can be an entry point for teenagers to create in game engines like Unity

BECCA ROSE

Becca is a learning manager on the Code Club projects team at the Raspberry Pi Foundation. She is a maker and educator and enjoys teaching and learning computing with arts and crafts, and anything related to craft generally!

> FOR YOUNG PEOPLE, THE FORMAT ENCOURAGES A CREATIVE FLURRY IN A SHORT SPACE OF TIME

people make games about the things that are around them and the things they dream about: sparkly unicorn poo, llamas, anxieties, favourite foods.

And in the process of making those things, they learn about game logic, movement, events, and interactivity, not because those things were taught explicitly, but because the making required it. The computing learning arrives differently, through the back door, carried in by something creators care about. When expression comes first, the technical becomes a tool, and it can be easier to think about what code is doing when it is serving something you want to share.

There's also something important about how flatgames don't need to be finished. They can be unresolved, unpolished, and a bit strange. There is permission to make something incomplete and still have it count, and this turns out to be quite generative. In the flatgame about the bee sting, the last scene requires the player to "press enter to restart". The point is not to win, but to make something else.

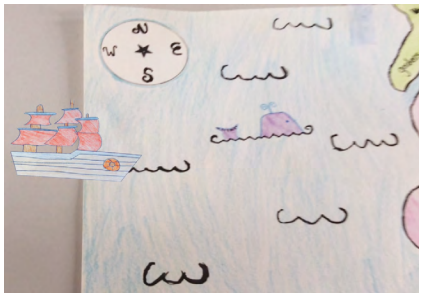
How to make a flatgame

A flatgame jam, which is a short, focused making session with a loose brief, works well either as a one-off event or as part of a longer project. Give people time to think about what they want to make before they start building. The making goes better when it starts with something personal. Even if you're not planning to work with physical materials in the final game, starting with pen and paper and getting crafty helps with the starting point.

We have a flatgame project (**helloworld.cc/flatgame**) you can try with younger creators. It walks the user through building a simple world using sprites and backdrops, and leaves plenty of room for young people to bring in their own drawings and photos. This involves some physical crafting and making, then bringing drawings or textures into Scratch. If the workflow from physical to digital feels like a barrier, you can also draw directly in the paint editor or gather images online. A simpler starting point for newcomers to Scratch could be more of an interactive drawing, for example adapting the flower project at **helloworld.cc/flower-generator**.

For older creators, Unity and Construct are powerful free tools for building 2D worlds, and the flatgame community has produced accessible tutorials (**helloworld.cc/flatjam2018**). Flatgames can be a brilliant entry point to Unity projects (**helloworld.cc/unity-guide**), as having a much narrower brief allows students to focus on making something creative with an otherwise complex tool — but hopefully inspiration will strike them a little less literally than in my case! (HW)

Apropolis is a 2D Unity flatgame



Younger creators can also make flatgames in Scratch

LEARNING TO TYPE WITH JOKES

Sean McManus explains how he used an AI tool to build a typing tutor called HaHaKeys where the punchline is the reward

What do John the Baptist and Winnie the Pooh have in common? They have the same middle name.

That's just one of the 200 jokes in HaHaKeys (**HaHaKeys.com**), a free website I developed to help Key Stage 2 children (aged 7–11) learn to type.

So far in my Code Club, we've been working in Scratch, but there is interest in advancing to Python and learning how to build websites with HTML and CSS. However, a recent Scratch project using `say` blocks illuminated the huge difference in typing skills across Key Stage 2.

I wanted to explore how we might improve these typing skills before we attempted text-based languages, which can be frustrating for children with limited keyboard skills.

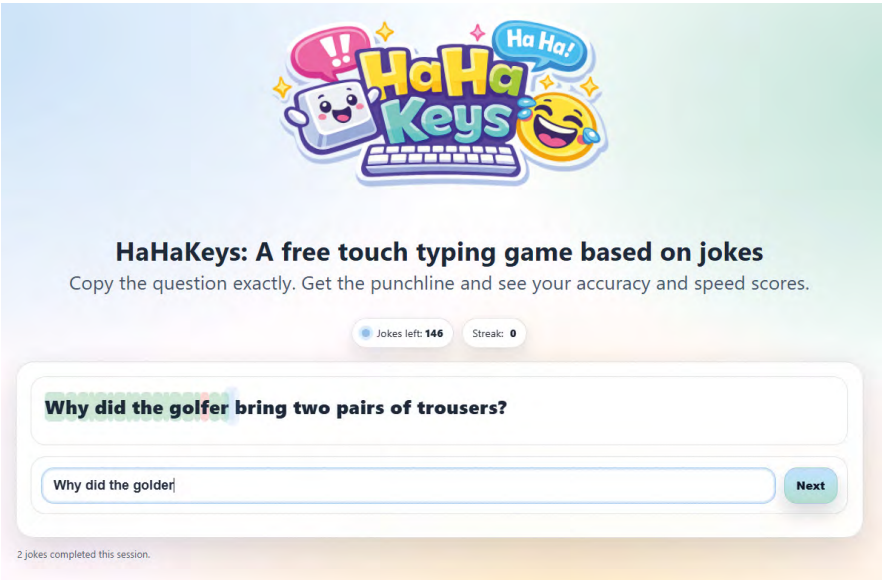
Removing the 'bored' from 'keyboard'

I had seen a number of typing tutors before, and seen children struggle to maintain their interest in them. These tools can be good for teaching finger positions, but they typically require the player to type a nonsense sequence of words. There might be an animation, such as a car in a race. Highly motivated children can do well with tools like this, but the content is not very enticing. The words are random, and the visual sequences are unrelated special effects in which the novelty fades fast.

When I asked my Code Club for their thoughts about typing tutors, only three of the six who had tried one thought it was any good.

I started to think about how I could make typing practice inherently rewarding by using more meaningful and entertaining text. My idea was to create a tutor in which the user types in the feed line for a joke and is rewarded with a punchline. For each line the user types, there's an incentive because they want to see the punchline, and a reward because they get to enjoy the joke.

At the same time, the program scores the speed and accuracy with which they enter the feed line, so they can track their progress.


Ha Ha!
HaHa Keys
HaHaKeys: A free touch typing game based on jokes
Copy the question exactly. Get the punchline and see your accuracy and speed scores.
Jokes left: 146
Streak: 0
Why did the golfer bring two pairs of trousers?
Why did the golder
Next
2 jokes completed this session.

Players are scored for their speed and accuracy as they type the feed line for a joke

Vibe coding a typing tutor

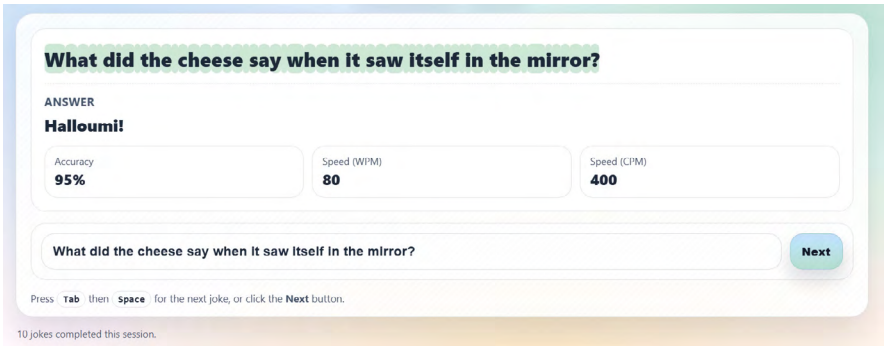
Although I have some reservations about the use of AI, it was the ideal way for me to make this app once I'd had the idea. I didn't have the time to hand-code it, and I didn't have the budget to pay someone else to do it, either.

You might have heard of vibe coding, which was Collins' word of the year in 2025. It describes the process of using AI tools to create software. You start by prompting the tool with a description of the app, and it gives you the code. You test the code, tell the tool what didn't work, and it gives you an updated version. Iterating in this way, you can create software relatively quickly without needing to look at the code.

In theory, vibe coding makes app development accessible to everyone. In practice, I've found that my coding skills have often been needed to guide the AI towards a solution, and to make the final edits to complete an app.

My initial prompt for ChatGPT was 250 words long, and included instructions to make the program highlight the letters as the user types them, measure the player's accuracy and typing speed, and include

When a player types the feed line correctly, they see the punchline as well as their accuracy and speed scores


What did the cheese say when it saw itself in the mirror?
ANSWER
Halloumi!
Accuracy
95%
Speed (WPM)
80
Speed (CPM)
400
What did the cheese say when it saw itself in the mirror?
Next
Press Tab then space for the next joke, or click the Next button.
10 jokes completed this session.

> WE CAN MAKE TYPING PRACTICE REWARDING BY USING MEANINGFUL AND ENTERTAINING TEXT

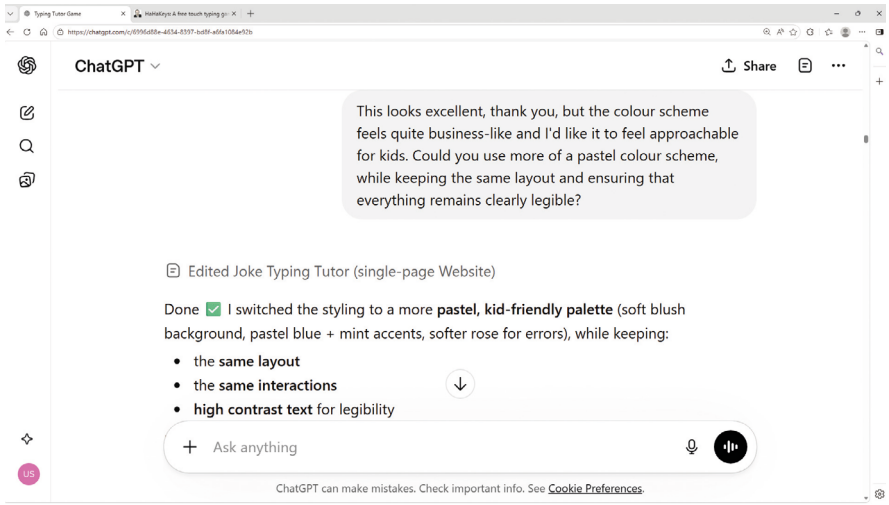
family-friendly jokes that don't repeat until they have all been played. My chosen format was a single-page website, because it's easy to use in the classroom without needing any set-up.

It took 20 ChatGPT prompts to arrive at the final version. Some of them were fixing bugs, some were adding features, and others were enhancing the layout and design from the rather bland businesslike starting point.

Part of the process of vibe coding is testing with a range of inputs, to be confident the program works as expected. Scoring didn't work well at first. For example, the accuracy was always shown as 100 percent, because the text had to be correct in order for the punchline to be revealed.

"Only use jokes you are confident are real jokes that are funny," I prompted. I didn't want the computer to try to make up jokes. I kept some of the jokes provided by the AI tool, which were familiar classics, and added many more of my own favourites, plus some suggestions from my Code Club. I have checked all the jokes myself to ensure they are appealing and suitable, and I add new ones as I come across them.

My son soon discovered that he could get an astronomical speed score by copying and pasting the feed line instead of retyping it. At first, I left this as a fun feature for kids to discover, but it proved too distracting in my Code Club, so I disabled pasting.


ChatGPT
Share
This looks excellent, thank you, but the colour scheme feels quite business-like and I'd like it to feel approachable for kids. Could you use more of a pastel colour scheme, while keeping the same layout and ensuring that everything remains clearly legible?
Edited Joke Typing Tutor (single-page Website)
Done I switched the styling to a more pastel, kid-friendly palette (soft blush background, pastel blue + mint accents, softer rose for errors), while keeping:
the same layout
the same interactions
high contrast text for legibility
Ask anything
ChatGPT can make mistakes. Check important info. See Cookie Preferences.

It took 20 ChatGPT prompts to create and refine the design for HaHaKeys

Testing HaHaKeys

I launched the app at **HaHaKeys.com** and we tested it in a Code Club session, where it went down well. Some of the children played it until the end of the session and competed with their neighbours to get the highest scores.

"I like that you get jokes and it's fun," said one boy. "The best thing is that it teaches you how not to make mistakes." He achieved an accuracy streak of 14.

Another said, "I like that the text is short and that you get a speed score." I noticed there was a natural rhythm to playing the game, because the clock stops when the punchline is revealed. Players have a tiny rest while they read the punchline and go into each new round fresh.

When I asked for a rating out of 10, the children all gave it 10 out of 10 or a crazy higher score. Some went on to play it at home, and some asked to use it in later Code Club sessions.

HaHaKeys is available now for anyone to use in their classrooms, Code Clubs, or homes. It works in the browser and has been designed for instant play, with no need for registration. It offers a fun additional way to develop keyboard skills, alongside other typing tutors. (HW)

SEAN MCMANUS

Sean is a Code Club volunteer and author of inspiring coding books including *Scratch Programming in Easy Steps*, *Mission Python*, and *Web Design in Easy Steps*. Find free chapters, Scratch assets, and more at **sean.co.uk**.

YOUR GUIDE TO PD REQUIREMENTS FOR COMPUTING EDUCATORS

Why professional development matters

Digital technologies shape almost every aspect of modern life, from how we communicate and learn to how we work and solve problems. Computing education helps young people develop the knowledge, skills, and confidence they need to participate in an increasingly digital world and to access future opportunities in education and employment. As demand for digital skills continues to grow globally, ensuring that all young people receive a high-quality computing education is more important than ever (helloworld.cc/royal-society; helloworld.cc/kemp).

At the heart of that education are teachers. Research consistently shows that teaching quality is the most important in-school factor influencing student outcomes, making educators one of the most valuable resources in any education system (**tdtrust.org**). Supporting teachers in their professional development is therefore not just beneficial for educators themselves; it is essential for improving outcomes for learners.

High-quality, sustained professional development (PD) helps teachers stay up to date with new technologies, strengthen their teaching practice, and build confidence in the classroom. It also creates opportunities to explore new approaches, connect with peers, and better support young people to develop essential digital skills. Evidence shows that investing in teacher development is also one of the most cost-effective ways to improve learner outcomes. Research from the Education Policy Institute in England found that high-quality PD can deliver the equivalent of an additional month of pupil progress each year and offers better value for money than many commonly used school improvement interventions, including performance-related pay, one-to-one tutoring, and extending the school day (**helloworld.cc/fletcher-wood**).

Computing education continues to evolve

Across many countries, governments recognise that digital and computing skills are increasingly important for participation in modern society and the future workforce. In England, for example, researchers have highlighted concerns about declining participation in computing qualifications, particularly among girls, and the need for stronger support for teachers through subject-specific professional development (**helloworld.cc/kemp**).

The same research found that many computing teachers reported feeling underprepared to teach parts of the curriculum and described access to continuous professional development as

National guidance in India recommends educators complete at least 50 hours of PD a year

© WESTOCK/stock.adobe.com

FURTHER READING

Fletcher-Wood, H. & Zuccollo, J. (2020). *The effects of high-quality professional development on teachers and students.* Education Policy Institute. (**helloworld.cc/fletcher-wood**)

Kemp, P., Wong, B., Hamer, J. & Copsey-Blake, M. (2024). *The future of computing education: Considerations for policy, curriculum and practice.* King's College London, University of Reading, and Nuffield Foundation. (**helloworld.cc/kemp**)

The Royal Society. (2025). *System upgrade required: Creating opportunities in computing education.* (**helloworld.cc/royal-society**)

Teacher Development Trust. (2025). *Teacher development: The CPD landscape in 2025.* Teacher Development Trust. (**tdtrust.org**)

Country	Current PD expectation
United Kingdom	England, Wales, Northern Ireland: schools are expected to provide PD, but there is no statutory minimum annual number of hours. Scotland: registered teachers are expected to complete 35 hours of professional learning annually through the Professional Update process.
Ireland	Teachers are expected to engage in ongoing professional learning as part of professional practice, but there is no nationally mandated annual PD-hour requirement for registration renewal.
United States	Requirements vary by state. Most states require professional learning for licence renewal, commonly ranging from 50–180+ hours over a 3–5-year cycle.
Kenya	Teachers participate in Teacher Professional Development (TPD) programmes accredited by the Teachers Service Commission (TSC) and aligned with the Kenya Professional Teaching Standards (KePTS).
South Africa	Teachers participate in the South African Council for Educators (SACE) Continuing Professional Teacher Development (CPTD) system, earning professional development points over 3-year cycles.
India	National guidance from National Council of Educational Research (NCERT) and Central Board of Education (CBSE) recommends at least 50 hours of PD a year for teachers and school leaders.

> “ TEACHER DEVELOPMENT IS A COST-EFFECTIVE WAY OF IMPROVING LEARNER OUTCOMES

inconsistent. Teachers and school leaders also emphasised the importance of ongoing training in both subject knowledge and inclusive teaching practices.

More recently, the Royal Society highlighted the growing demand for digital skills and the need to strengthen computing education pathways to help address workforce shortages and broaden participation in computing-related careers (**helloworld.cc/royal-society**).

Professional development expectations around the world

Professional development requirements vary significantly between countries. In some education systems, teachers are expected to complete a set number of PD hours each year. In others, professional learning is required but the number of hours is left to schools, employers, or professional bodies.

While formal requirements differ, the overall expectation is similar: educators should continue developing their knowledge and skills throughout their careers.

Research from the Teacher Development Trust found that most teachers and school leaders in England participate in some form of professional development each year, although access, time, and funding remain significant challenges (**tdtrust.org**).

Making professional learning more accessible

Finding time for professional development is not always easy. Many educators balance busy teaching schedules, administrative responsibilities, and other competing priorities. Funding constraints and device availability can also limit access to external training opportunities.

That is why flexible, high-quality professional learning is increasingly important. Online and self-paced learning opportunities can help educators develop their computing knowledge when it suits them, while also guiding them on practical classroom application and connecting them with professional communities.

The Raspberry Pi Foundation's Training Hub was created with both learner outcomes and these challenges in mind (**rpf.io/trainonline**). It offers free, self-paced courses designed specifically for educators worldwide, wherever they are in their computing education journey. The courses are written by expert educators and cover topics including Python, Scratch, AI and machine learning, cybersecurity, and computing pedagogy. Educators can also connect with peers, share ideas, and receive certificates that recognise their learning.

Whether you are meeting formal PD requirements or simply looking to build your confidence in teaching computing, professional development can help you stay current, strengthen your practice, and support your students to thrive in today's digital world. (HW)

RIK CROSS

Rik is director of professional development at the Raspberry Pi Foundation. He enjoys education and game development and projects that combine these two interests.

BIOBOTS IN COLOMBIA PROGRAMA

Teaching computational thinking without computers

Colombia faces a stark digital divide between its urban and rural areas. According to the National Administrative Department of Statistics, only 22.7 percent of rural schools had internet access in 2021, compared to 90.8 percent of urban schools (helloworld.cc/dane-edu). Geography and historic conflict have further deepened this divide. Yet the demand for digital skills is urgent: a 2025 forecast shows a shortfall of about 112,000 software developers (helloworld.cc/colombia-dev-gap). BioBots aims to be part of the solution, bringing computational thinking to underserved areas through a method that is accessible, affordable, and inclusive. It is part of the nationwide programme Colombia Programa (or 'Colombia Codes' in English, helloworld.cc/colombia-programa) and originates from an agreement signed between the Colombian Ministry of Information and Communication Technologies and the British Council.

Framework and rollout

Launched in December 2023, Colombia Programa is inspired by England's National Centre for Computing Education model and integrates coding hubs, peer mentorship, and teacher training. It includes 79 validated curricular guides for kindergarten to grade 11 (ages 4 to 17) and incorporates unplugged tools like BioBots alongside high-tech options such as the micro:bit.

Its rural-focused strategy includes health-style deployments and mobile outreach, ensuring communities with limited access to devices and connectivity can still benefit from quality STEM education.

© Colombia Programa
Escenario 3 – El nevado se derrite
Escenario 2 – Deforestación en el Chocó
Escenario 1 – Exploremos la Sierra Nevada

Educators trying out the BioBots game during teacher training

What is BioBots?

BioBots (**helloworld.cc/biobots**) is an unplugged educational board game designed for kindergarten age and up that uses character tiles, route-setting cards, puzzle boards, and instructional sequences. Players learn to create and debug instructions, recognise patterns, and develop algorithmic thinking — all without the use of screens. The game scaffolds learning by increasing complexity across 14 scenarios, promoting six core computational subskills — algorithmic thinking, debugging, decomposition, pattern recognition, abstraction, and logical thinking. It is a problem-solving engine, which means that, unlike a traditional game, it has a rule-based architecture that allows for the creation of multiple scenarios infinitely, thanks to a modular system and the use of procedural random generation. As a result, it can be played countless times in different ways.

For example, one mission is to rescue a toucan that is placed on a terrain tile ahead of your BioBot. Using the program tiles, you choose three commands to create an algorithm to reach the toucan. You can create different terrain maps, add challenges, and create your own missions (**helloworld.cc/biobots-gameplay**).

The design of BioBots reflects Colombian geography and environmental values, ensuring cultural relevance while fostering computational skills through environmental care tasks. The game emphasises a stage-based cognitive approach, supporting learners through increasingly complex tasks.

The unplugged format of BioBots supports flexible implementation across various learning contexts

Each scenario reinforces a mix of logic, abstraction, decomposition, and debugging — the foundations of computational thinking (**helloworld.cc/colombia-pedagogy-resources**).

Maryuri Agudelo, a BioBots mentor working with rural teachers, said: "I value the strength and courage of rural teachers, who face great challenges to ensure learning. Strategies like BioBots are key, as they do not require connectivity or advanced technological resources. They allow for analogue work, awakening in teachers and students a sense of wonder and motivation that transforms teaching. Seeing teachers play, contextualise narratives, and take on new challenges with creativity shows the impact of connecting education with local communities."

Impact and recognition

BioBots won two GamiCon48V awards in April 2025: Best Overall Use of Gamification and People's Choice Award for Best Low-Tech Gamification (**helloworld.cc/biobots-yt**). Educators have reported higher engagement and skill development. For example, primary students who played the game in 2024 demonstrated a statistically significant increase in their mean overall computational thinking scores between pre- and post-tests during the pilot phase.

> "THIS ISN'T JUST A GAME; IT'S A TOOL TO TRANSFORM HOW TEACHERS TEACH

Colombia's interim Minister of Information and Communication Technologies, Belfor Fabio García, described BioBots as "a powerful tool to spark interest in problem-solving and logical thinking — essential 21st-century skills". British Council Country Director for Colombia Felipe Villar Stein added, "Gameplay can close gaps and open worlds of possibilities for girls and boys across Colombia."

Javier Velásquez, one of the game designers, highlighted the transformational power of BioBots: "This isn't just a game; it's a tool to transform how teachers teach and students learn."

Scalable, sustainable, culturally relevant

BioBots is designed for replication. All curricular materials are open-licence, enabling easy adaptation and localisation. Printable boards, translated guides, and 3D-printable parts are available for educators worldwide. The modular structure supports flexible implementation across various age groups and learning contexts.

The game's development is rooted in equity, gender inclusion, and ecological awareness. Its low-tech format is ideal for underserved regions, ensuring that learning is not limited by device access or internet availability. At the same time, it can serve as a practical resource for urban schools and teachers who want to develop learners' computational thinking through a combination of plugged and unplugged approaches.

Melany, a third-grade student from Puracé-Cauca, noted: "[Playing BioBots] has been a very lovely experience; I have learnt about several subjects and how to be much more responsible. Now my classroom is more fun, and we can learn by playing without being connected."

What this means globally

BioBots offers a compelling model for equitable, low-resource, human-centred computing education. It exemplifies

how unplugged strategies can bring computational thinking to communities with limited access to technology. The model is replicable, scalable, and culturally responsive.

For countries facing similar infrastructure gaps, BioBots is a blueprint for change, underscoring a powerful truth: computational thinking is not about machines — it's about mindset.

Practical advice for adoption

Teachers and advisers seeking to bring BioBots or similar unplugged tools into their context should consider the following tips.

- **Consult the pedagogical material (helloworld.cc/biobots)**: BioBots comes with guides that will help you understand how it works and the rules to follow in each scenario
- **Start with teacher training**: provide hands-on workshops and mentoring to build confidence and familiarity
- **Create collaborative learning networks**: establish coding hubs in rural, suburban, and metropolitan areas to encourage peer support
- **Sequence lesson progression**: begin with simple puzzles, reinforce logical structure, and then add abstraction and debugging
- **Embed effective education**: use gameplay to promote emotional intelligence, teamwork, collaboration, resilience, and group reflection
- **Adapt and localise content**: use local references, flora/fauna, and culturally relevant contexts to boost engagement

Next steps

BioBots will continue to scale nationwide across the Andean, Orinoquía, Amazonia, and Insular regions as we provide professional training for 700 teachers across 300 rural schools to directly benefit approximately 14,000 students. The British Council will also be ready to share the results of the impact evaluation that is currently being conducted with 13 educational institutions, 17 teachers, and 446 students. By the end of 2026, Colombia Programa, the project that produced BioBots, aims to train 11,200 teachers and reach nearly 900,000 students across 420 coding hubs. Other countries, such as Jamaica and Brazil, have expressed interest in adapting this hub model.

Start where you are

BioBots shows that meaningful tech education doesn't require technology. With an engaging board game, dedicated teachers, peer communities, and government backing, the Colombia Programa programme is making computational thinking accessible, inclusive, and human-centred.

For teachers in low-resource settings around the globe, BioBots sets a clear example: start where you are, think like a programmer, and empower every student to solve tomorrow's problems. And if resources are not an issue for your school, classroom, or students, no problem — BioBots can also be a great supplement to introduce computational thinking effectively while having some fun with your learners too. As Juan Diego, a student from Sotará-Cauca, told us: "BioBots is a cool game that teaches to code, to learn and to think." (HW)

Educators and students trying out the BioBots game during a teacher training session

MARCELA ORTIZ-CARDONA

Marcela is a PhD candidate at the University of Huelva, performing research in didactics of experimental, social, and mathematical sciences. She is also a research scholar at the STEM Center for Outreach, Research & Education at Texas Tech University and a national technical lead in computational thinking for rural zones of Colombia Programa.

LORENA MOLINA-CASTRO

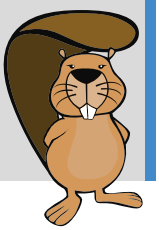
Lorena is a computer engineer with postgraduate studies in digital pedagogy. With over 25 years' teaching experience, she has worked across a wide range of levels, from preschool to master's programmes. Over the past 15 years she has also worked as an academic consultant, instructional designer, and teacher trainer on projects led by the British Council.

THE BEBRAS PUZZLE PAGE

Each issue, **Andrew Csizmadia** shares a computational thinking problem for your students based on the work produced by the International Bebras Community

ABOUT BEBRAS

Bebras is organised in over 90 countries and aims to get students excited about computing and computational thinking. Last November, 526,000 students participated in the UK annual challenge. Our archived questions let you create your own automarking quizzes at any time during the year. To find out more and to register your school, head to **bebras.uk**.



DOMAIN

Algorithms, logic, and programming

SKILLS

Algorithms and evaluation

AGE

6–10 years

DIFFICULTY RATING

Age 6–8 medium
Age 8–10 easy

THE PROBLEM: HONEY POT

Bertie the bee has to fill the honey pot as quickly as he can.

Bertie can open and close the taps in the pipes below by clicking on them. By doing this, he can control which way the honey flows.

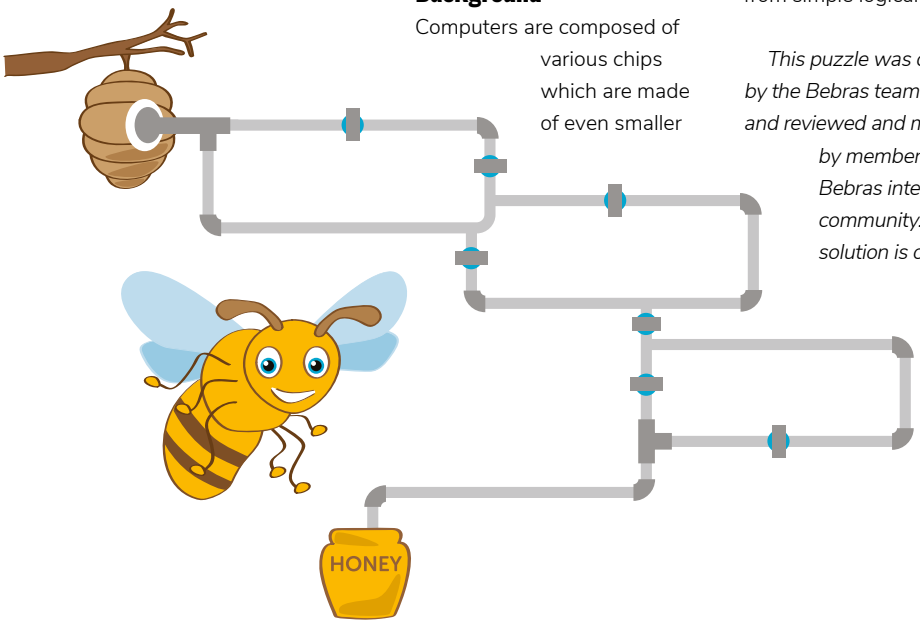
Task

Which taps must Bertie open to make the honey reach the pot by the shortest possible path?

Background

Computers are composed of various chips which are made of even smaller parts: electronic circuits. Electronic circuits are in turn composed of logic gates. Logic gates act like valves, except that instead of water they conduct electricity, and instead of pipes they have wires. This means that our modern electronic devices (including complex ones like computers and smartphones) are built up from simple logical operations.

This puzzle was developed by the Bebras team in Australia, and reviewed and modified by members of the Bebras international community. The solution is on page 77.



KEYWORD SPOTLIGHT: SHORTEST PATH

Defining everyday words and phrases in computer science

The concept of a shortest path comes from graph theory and is widely used in computing. At its heart, it answers a simple question: What is the most efficient way to get from one point to another?

A helpful way to introduce this to younger pupils is through familiar contexts. For example, imagine a map showing several towns connected by roads. If you want to travel from Town A to Town D, there may be multiple routes. Some are longer, some shorter. The shortest path is the route with the smallest total distance.

For older students, we often represent problems like this using a graph. A graph is made up of:

- Nodes (or vertices): the points (e.g. towns, bus stops, webpages)
- Edges: the connections between them (e.g. roads or links)
- Weights (optional): numbers that show cost, distance, or time

The shortest path is the route between two nodes where the total weight of the edges is as small as possible. (HW)

MAKING STEM OUTREACH WORK IN BUSY SCHOOLS

Alan O'Donohoe shares practical tips on how to reduce friction in STEM outreach

STEM outreach in schools often fails for surprisingly simple reasons. Schools are busy places. If an activity requires software installations, complicated timetabling, unreliable Wi-Fi, or endless email chains, even the most exciting workshop can slide to the bottom of the priority list.

Over the last three academic years, our outreach team has delivered hundreds of funded STEM engagement sessions in schools across Priority Education Investment Areas (PEIAs) and other communities traditionally underrepresented in STEM across the UK. During our outreach, I've become increasingly interested in one question: How do we make STEM outreach genuinely easy for schools to say yes to? The answer, I believe, lies in reducing friction wherever possible — for teachers, for students, and for the organisations delivering the sessions.

The problem with 'more'

One of the biggest lessons we learnt early on was that outreach is not always improved by reaching the largest possible audience. Schools understandably want opportunities to benefit as many pupils as possible. However, I often found myself using what I now call the cake analogy. If you cut a cake into ten slices, everyone gets something meaningful. If you divide the same cake into 200 tiny pieces, nobody leaves satisfied.

STEM outreach can work in the same way. Smaller groups, longer sessions, and deeper conversations often create far more impact than trying to deliver something brief to an entire year group.

This became particularly important when working with students who may already feel excluded from STEM pathways — including girls, students with SEND, or pupils from disadvantaged backgrounds. A quieter environment, more discussion time, and opportunities to contribute without pressure can completely change how a young person experiences computing or engineering.

Reducing friction for teachers

Teachers can be incredibly skilled at spotting the difference between a genuinely useful opportunity and an activity that creates additional workload. That means the practical details matter enormously.

Over time, we started simplifying everything we could:

- No software installations
- No account creation
- No dependency on school Wi-Fi
- Minimal classroom set-up
- Adaptable activities that could work in almost any room

The more 'plug and play' the experience became, the easier it was for schools to participate in it.

One of the most useful changes we made was relying far less on specialist hardware than people might expect. In fact, some of the strongest engagement came from unplugged activities. Students investigated mysterious 'sonic tins', solved physical logic challenges, analysed patterns, and cracked combination locks. These activities often generated richer conversations than highly technical set-ups because they removed the fear of 'getting the technology wrong'.

For some students, particularly those lacking confidence, a physical puzzle can feel much more approachable than a screen full of code.

Why timing matters

Another practical takeaway involved session length. Many enrichment activities are squeezed into 45-minute or one-hour slots because that fits neatly into timetables. The challenge is that meaningful discussion often disappears first when time becomes tight. We found that longer sessions created space for reflection, teamwork, and career conversations — the parts students often remembered most afterwards.

Students rarely remember every technical detail from a workshop. What they do remember is:

- Whether they felt capable
- Whether their ideas were valued
- Whether they could imagine themselves belonging in STEM

That takes time. Longer sessions also reduced the pressure on delivery staff. Instead of rushing through activities, facilitators could

pause, adapt explanations, and respond to unexpected questions without feeling they were constantly chasing the clock.

Making participation visible

One challenge in outreach work is that confident students naturally dominate discussions. Like many teachers, I wanted quieter students to feel just as involved.

We experimented with simple participation systems, including raffle tickets or small rewards for thoughtful contributions, teamwork, resilience, or kindness. Interestingly, this often shifted classroom dynamics quite quickly. Students who rarely volunteered began contributing because the environment felt safer and less competitive.

We also used low-tech assessment tools such as Plickers cards (**plickers.com**) to gather student responses anonymously. This proved especially useful when discussing confidence, aspirations, or misconceptions around STEM careers.

Privacy matters more than we sometimes realise. A student may be reluctant to admit they do not understand something — or that they have never considered a STEM career — if their peers can immediately see their answer. Simple systems can create much more honest conversations.

The importance of relevance

One of the easiest ways to lose students during outreach is to present STEM careers as abstract or distant. Students engage far more readily when they can see a clear connection between classroom activities and real-world roles. That does not necessarily mean overwhelming them with technical information. In fact, the most effective conversations were often the simplest:

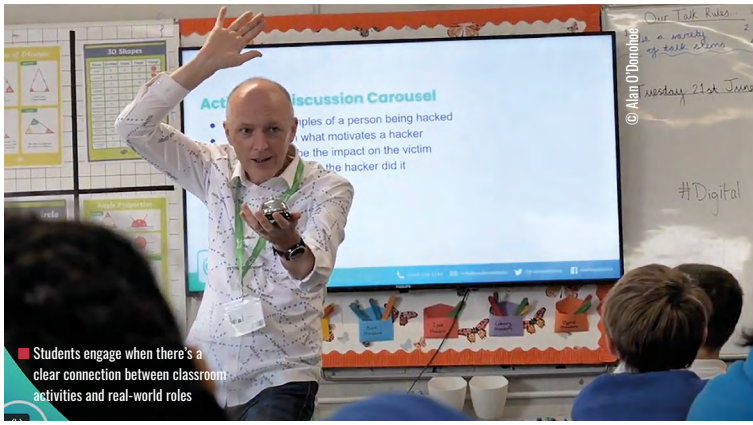
- Who designed this?
- Who tested it?
- Who fixed it?
- Who communicated the idea?
- Who worked as part of the team?

These discussions helped students recognise that STEM is not a narrow pathway reserved for a small group of clever people. Modern STEM industries need designers, communicators, technicians, engineers, analysts, project managers, and creative thinkers. For some students, simply realising that there is more than one route into STEM can be transformative.

Outreach is a partnership

If there is one final lesson I would like to share, it is this: schools are not venues. The most successful outreach happens when schools are treated as genuine partners rather than audiences.

Teachers understand their students better than any visitor ever could. The most effective sessions were always shaped by conversations with staff beforehand, and those conversations often mattered more than the workshop itself.



Students engage when there's a clear connection between classroom activities and real-world roles

- Which students need encouragement?
- Which topics connect well with current learning?
- What barriers exist in this particular community?
- What has worked — or failed — previously?

I also learnt that persistence matters. Many schools are understandably cautious about externally funded programmes because they have experienced initiatives that overpromised and underdelivered.

Sometimes, trust is built slowly. A thoughtful email, a short conversation with a computing lead, or a recommendation from another school can make all the difference.

Start simple

For anyone thinking about developing their own outreach programme, my advice would be to start smaller than you think you need to. You do not need expensive equipment, large teams, or complicated resources to create meaningful STEM engagement. Some of the most memorable moments I've seen involved paper prototypes, collaborative puzzles, role play, simple tools, or students discussing ideas around a classroom table. What matters most is reducing barriers to participation, barriers to confidence, and barriers to opportunity.

The most effective outreach programmes are the ones that respect teachers' time, create inclusive experiences, and help young people feel that STEM might have a place for them after all. And sometimes, that starts with something as simple as making it easier for a school to say yes. (HW)

ALAN O'DONOHOE

Alan has over 30 years' experience as a leader in education. On behalf of Exa Networks Ltd., he runs the Exa Foundation, where he currently leads a national, UK government-funded programme promoting STEM careers to disadvantaged and underrepresented 8- to 14-year-olds (**alan.odonohoe@exa.net.uk**).

© JackF/stock.adobe.com

PROMOTING HUMAN DECISIONS IN AN AGE OF AI

Tracy Gardner and **Michael Conterio** discuss how to teach the important skill of decision-making to young people

"A computer can never be held accountable, therefore a computer must never make a management decision" — IBM Training Manual, 1979 (helloworld.cc/ai-decision-line).

Just because AI systems can make decisions doesn't mean they should. We have already described the importance of keeping "a human in the loop" (**helloworld.cc/24**), but we don't just want humans to be checking the outputs of AI systems — we need humans who are capable, willing, or even enthusiastic to be making the kinds of decisions that we don't want to outsource to AI. How do we prepare young people for a future in which they are making those decisions?

Whether it's questions of ethics, impact (especially using systems thinking to identify potential wider implications), resource management (financial, human, and environmental), aesthetics and taste (cultural relevance), or quality, we need to support young people to recognise the importance of making some decisions themselves and to develop the values and expertise to make them.

FURTHER READING

'AI decision-making: Where do businesses draw the line?', IBM. **(helloworld.cc/ai-decision-line)**

'Researchers Asked LLMs for Strategic Advice. They Got "Trendslop" in Return', Harvard Business Review. **(helloworld.cc/trendslop)**

Demonstrating the need for decision-making

"Leadership is ultimately about making hard choices in conditions of uncertainty and taking responsibility for them. AI cannot and should not be a substitute" — Harvard Business Review (**helloworld.cc/trendslop**).

A good start, to demonstrate to students why they still need to make some decisions, is to demonstrate the limitations of LLMs and the risks of delegating decisions to them. A popular current example is asking an LLM, "I want to wash my car and the car wash is close. Should I drive there or walk?" LLMs frequently respond that you should walk. Most humans would quickly spot that this makes no sense. In practice, issues like this could be more subtle, such as missing or biased training data leading to problems in particular contexts or among particular groups of people. Students need to have strong critical-thinking skills to spot such problems, and not unthinkingly rely on AI responses.

You can then build on this by asking thought-provoking questions such as, "Would you want an AI system to decide what art should

go in an art gallery, or who should be given a job, or which party should be in government? Which AI system would you trust to do this? And if you are making decisions about AI systems' judgements, why couldn't you have made the decision yourself?"

There are also decisions involved in getting AI systems to generate outputs. AI systems are getting more and more capable of generating code. Increasingly, software developers are spending their time making decisions about what code should and should not exist, and reviewing code to decide whether it is good enough to be included in a product. No one wants to live in a world of AI slop, so how can we ensure that we create smaller amounts of high-quality software? What do your students think about this?

Similarly, we can generate vast quantities of images, text, and video for human consumption — whether with AI tools or not. But attention is limited and quality, ethics, cultural relevance, suitability, and aesthetics matter. How do we decide what to create or consume? Are your students aware that machine-learning-based algorithms are already directing their feeds?

This article is made of decisions. It is our specific take, based on conversations between two humans. We crafted our current opinions into an article that we hope is interesting to read and provokes further conversations. It's not an 'average of the internet', and by writing it we also developed our own thinking and practised our own decision-making.

Helping young people to develop decision-making skills

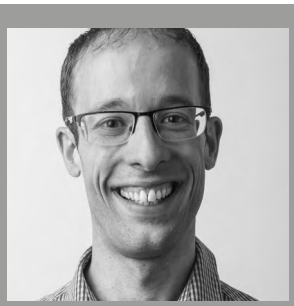
As with any skill, young people need to be given opportunities to put decision-making into practice in authentic situations, as well as to reflect (ideally with feedback) about their performance while practising again.

You can provide some of these opportunities by designing group work carefully. If different parts of the task need to be completed by different people (due to time and resource constraints), students need to communicate and collaborate on group decisions, but also have responsibility for making individual decisions in their area. Allow time for discussion on the impact of the decisions they make.

Learning to program is still valuable, but you should place more importance on understanding requirements and user needs (deciding what to create) and reading, evaluating, and testing (deciding whether it is good enough).

> "AS WITH ANY SKILL, YOUNG PEOPLE NEED OPPORTUNITIES TO PUT DECISION-MAKING INTO PRACTICE

And what about the bigger decisions, the ones about the wider effects of changes on society? Luckily, humans have long had a means of trying to think through these: through telling stories, both fictional and non-fictional. By helping students to consume, discuss, and critique a wide variety of media, we can help them not only to make decisions individually here and now, but to work together to decide the society and the future they want to create. (HW)



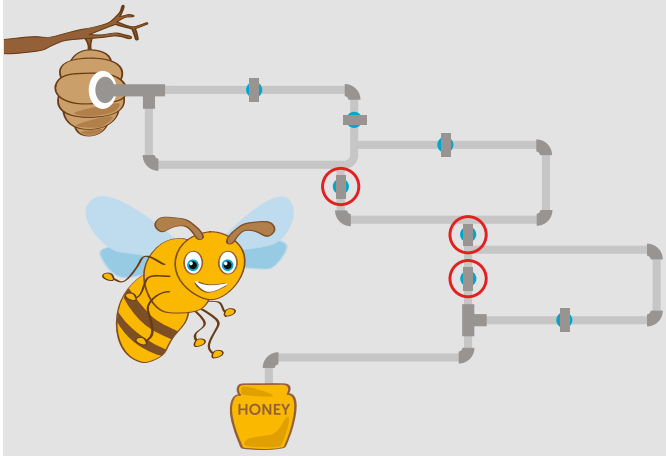
TRACY GARDNER & MICHAEL CONTERIO

Tracy is a computer scientist, tech industry professional, technology educator, and co-founder of Flip Computing. Michael is a former physicist and now works as an online course production manager at the Raspberry Pi Foundation.

BEBRAS PUZZLE

BEBRAS PUZZLE SOLUTION: HONEY POT (PAGE 73)

The image below shows the three taps that need to be opened to allow the honey to flow along the shortest path. All the other taps should be closed.



Explanation

This task requires the students to analyse a model and solve two problems:

- Find the shortest path in a network of pipes
- Work out how to arrange a series of valves to ensure this path is followed

REDEFINING WHAT A CODER LOOKS LIKE

Miami nonprofit Code/Art inspires girls to explore computer science careers using art, tech, and creative competition

"I never thought I could code — but creating my self-portrait changed how I see myself."

Those are the words of Rosina, a middle-school student from South Florida. Growing up in and out of foster homes, she often felt disconnected from her classmates and unsure of her abilities. That changed when her teacher introduced her to Code/Art's CodeYourSelf™ competition, which invites students to code a digital self-portrait.

At first, Rosina wasn't sure she could do it. But with her teacher's guidance, she completed her submission — and was stunned when she was selected as a finalist. For the first time, she felt recognised for her creativity and intelligence. That recognition sparked a new confidence, greater engagement in school, and an interest in computer science — a path she had never imagined before she discovered the creative possibilities of code.

Why art and code works

In today's world, art and technology are deeply connected, yet these links are often invisible to the casual observer. Coding is more than numbers and logic: it's a creative tool for designing, experimenting, and problem-solving. Code/Art (**code-art.com**) is a nonprofit dedicated to increasing the number of girls studying computer science, in part through competitions that make these connections visible, transforming coding into an approachable and expressive medium.

Instead of starting with abstract concepts, students create something personal and familiar: a self-portrait, an animation, or a game. This art-first approach lowers barriers and shows that coding can be fun, visual, and imaginative.

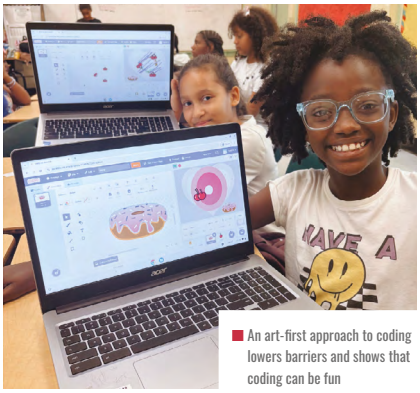
As one past participant, Alejandra, explains: "What I find most rewarding about this competition is the fact that it's challenging; the fact that you are putting your creativity to the test … and also being able to compete with girls who are amazing and seeing all their cool artwork!"

And while students are creating, they're also gaining transferable skills: critical thinking, collaboration, and digital fluency. For girls in particular, this blend of art and technology helps dismantle stereotypes about who belongs in computer science, replacing the old image of a nerdy coder with a new image of themselves as smart, creative, and more-than-capable coders.

WHY ART AND CODE BELONG TOGETHER

- **Creativity meets logic:** coding isn't just about numbers; it's also about imagination and design
- **21st-century skills:** projects build collaboration, problem-solving, and digital fluency
- **Empowering girls:** art-first coding redefines what a programmer looks like

Code/Art's CodeYourSelf™ competition invites students to code a digital self-portrait



An art-first approach to coding lowers barriers and shows that coding can be fun

THE CODE/ART TEAM

Lisa Hauser, Kaylla Torres, Romina Polo, and Amy Austin Renshaw create art-infused computer science curricula, programmes, and educator resources that help students, especially girls, discover the creative possibilities of technology. Since 2016, their competitions have inspired students to submit over 11,300 coded works of art.

> "I DIDN'T THINK IT WOULD CHANGE MY FUTURE ... BUT CODE/ART GAVE ME THE CONFIDENCE TO DREAM BIGGER

From first-time coder to future changemaker

Anne-Sophie, originally from Haiti and now pursuing a computer science degree at Williams College in Massachusetts, never imagined herself in technology when she was back in high school. "When my high-school math teacher introduced me to Code/Art's creative coding competitions, I didn't think it would change my future," she recalls. After coding her first self-portrait and being placed as a finalist in the CodeYourSelf™ competition, Anne-Sophie's confidence and ambition soared.

Encouraged by her teacher, she and her classmates entered another Code/Art competition, the Changemaker Challenge, and won first place for their website promoting financial literacy for secondary-school students. This experience deepened her passion for computer science and inspired her to pursue more opportunities to learn, lead, and give back. She went on to join a Code/Art club, founded her own nonprofit, HAITECH Learning (**haitechlearning.org**), to expand access to computer science education in Haiti, and ultimately earned an Amazon Future Engineer scholarship to study computer science at Williams. Anne-Sophie's journey illustrates how creative coding can be the spark that opens doors to new passions, careers, and ways of giving back.

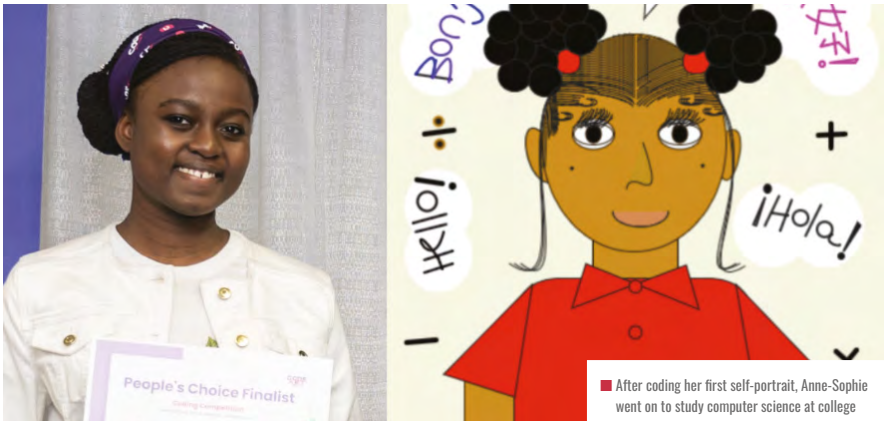
Building a community of creators

Beyond competitions, Code/Art works hard to ensure that students feel part of something bigger. Student creations are showcased through gallery-style exhibits at Code/Art Fest, Code/Art's annual conference for K–12 educators and students, as well as at school showcases, online features, and even in a printed coffee-table book featuring illustrations of their coded portraits. These public celebrations help participants see their work not only as school projects, but also as contributions to a growing community of young innovators.

Get involved

Get started now! Code/Art's annual competition is currently open, with entries due in February 2027. Whether you're a teacher looking to inspire your students, or a young person curious to try coding for the first time, this is your chance to discover what happens when creativity meets code. Visit **code-art.com/compete** to learn more about Code/Art's competitions.

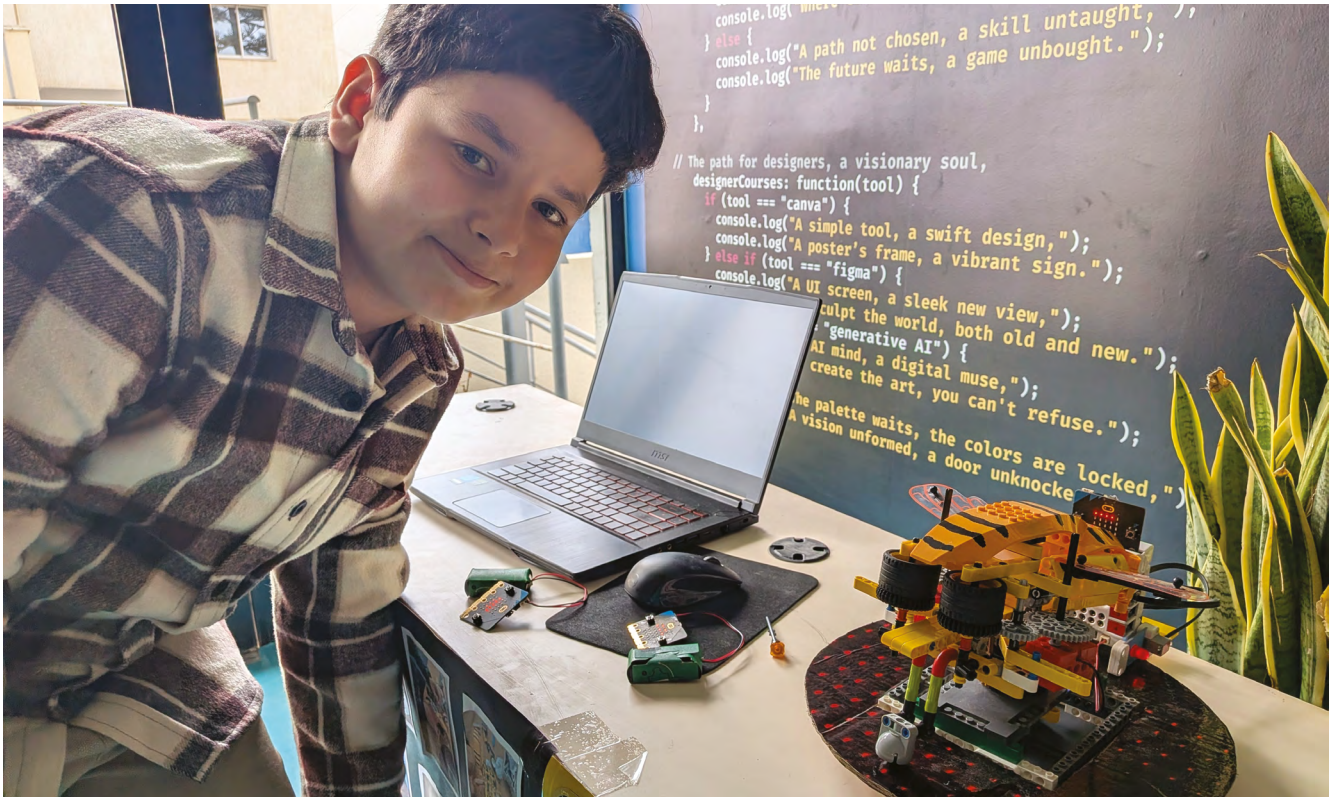
You can also access free video tutorials to guide your students (or yourself!) through the process by creating an account on our learning portal at **hub.code-art.com**. Together, we can redefine what a programmer looks like, one art piece at a time. (HW)



After coding her first self-portrait, Anne-Sophie went on to study computer science at college

HOW TEACHERS CAN GET INVOLVED IN ART + CODE EDUCATION

- Bring a Code/Art competition into your classroom
- Display students' coded art in hallways, media centres, or local galleries
- Encourage collaborative team projects in which students' creativity can shine
- Remind students that coding isn't just tech, it's self-expression


console.log("A path not chosen, a skill untaught,");
} else {
console.log("A path not chosen, a skill untaught,");
console.log("The future waits, a game unbought.");
}
},
// The path for designers, a visionary soul,
designerCourses: function(tool) {
if (tool === "canva") {
console.log("A simple tool, a swift design,");
console.log("A poster's frame, a vibrant sign.");
} else if (tool === "figma") {
console.log("A UI screen, a sleek new view,");
sculpt the world, both old and new.");
"generative AI") {
AI mind, a digital muse,");
create the art, you can't refuse.");
he palette waits, the colors are locked,")
A vision unformed, a door unknocke

BUILDING A MORE INCLUSIVE FUTURE IN TUNISIA

How DotSlash AUPTIMISME champions creativity

DotSlash AUPTIMISME (helloworld.cc/dotslash), a global Code Club partner, is an organisation whose mission is to combine inclusive technology education and creative computing, ensuring that all children in Tunisia have the opportunity to become digital creators.

Tarek Seghiri, founder of DotSlash AUPTIMISME, spoke with Ruhee Shah, global partnerships coordinator at the Raspberry Pi Foundation, about his unique school partnership model and how the organisation is a champion for creative freedom in the classroom.

"AUPTIMISME is a blend of two French words: "autisme" (autism) and "optimisme" (optimism) — and that name carries our whole story. It began as a project of inclusive technology education, born from the conviction that creative computing can transform the lives of all children — starting with those the system most often leaves behind." Tarek shared. Today, through CUBE, their 'Centre Technologique & Créatif', and a growing network of 13 partner schools, the organisation brings coding, digital making, and creative technology to young Tunisians.

Creating spaces for young people to thrive

For Tarek, computing education is far more than just a modern technical skill; it is an important tool for equity and personal expression. This is especially true for the diverse groups of learners his clubs welcome each week. "Computing is the new literacy — and for some children, it is more than that: it is a voice. For a non-verbal child, a Scratch project can be a way to express a story, a joke, a feeling. In Tunisia, talent is everywhere but opportunity is not."

To make these opportunities as accessible as possible, DotSlash AUPTIMISME utilises a school partnership model. Currently, they operate seven clubs across the semi-rural Cap Bon region and five clubs across Greater Tunis.

By embedding the clubs directly inside the schools the children already attend, they remove the logistical, administrative, and trust barriers that often hinder extracurricular activities in the region. The schools provide the safe framework, while DotSlash AUPTIMISME trains the mentors, who are a mix of school teachers and local university students, and manages the technology. This familiar setting has made it very easy for families to engage, helping young coders learn together.

The freedom to create

DotSlash AUPTIMISME is committed to learner autonomy. In these clubs, children are completely free to choose their own projects, guided by their personal interests rather than a rigid curriculum.

Tarek initially faced some obstacles — parents expected a traditional classroom set-up, schools focused on strict form, and even eager mentors wanted to step in too quickly. However, sticking to his approach has had incredible results. Attendance and engagement soared because the learning was driven by the children's desire to build things they genuinely cared about.

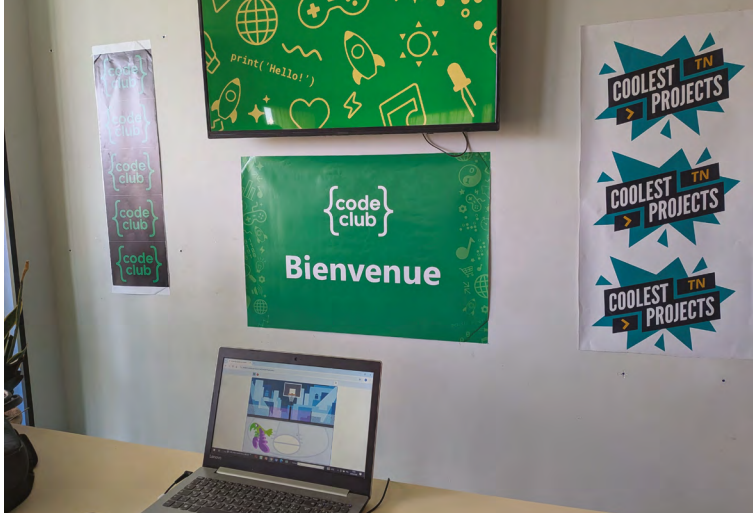
> “A CULTURE OF CELEBRATING INCOMPLETE OR BROKEN CODE STRIPS AWAY THE FEAR OF FAILURE

Instead of an inflexible curriculum, structure comes through supportive mentoring. Code Club's step-by-step projects act as a toolbox to kickstart ideas and teach core skills, but the final project is always up to the creator.

This creative freedom is taking on exciting new shapes across their network. At the Almaarifa School in Ennasr, Ariana, the school's leadership took the step of integrating coding directly into the primary school curriculum. Every single pupil now receives an hour of weekly coding sessions with end-of-term assessments, turning the classroom into a collaborative space where students and teachers learn the language of technology together.

Celebrating every step

Every DotSlash AUPTISIME Code Club session wraps up with a 'demo moment'. Creators stand up to showcase what they have built, even if the project is incomplete or the code is currently broken. By celebrating the attempt or a clever bit of debugging, the clubs actively strip away the fear of failure.



This culture of celebration reached a massive milestone with the launch of the first Coolest Projects Tunisia event. Bringing together pupils from the Cap Bon region, young creators proudly showcased 28 unique projects on stage to an audience of peers and parents. Tarek said, “Parents came along, and seeing their children present their own creations changed the way many of them look at what their kids are capable of. For children from a semi-rural area, standing up and showcasing a project they chose, designed, and built themselves is a powerful experience — exactly the confidence we are working to build.”

What's next for DotSlash AUPTIMISME?

When young creators eventually move on from the clubs, Tarek hopes they leave with far more than just technical literacy. The ultimate goal is to instil a deep sense of confidence, giving them the lifelong reflex to look at a difficult problem and say “I can figure this out,” rather than “I can't do this”.

Looking towards the remainder of the year, DotSlash AUPTIMISME is focused on two major objectives. The team will be expanding their school collaborations, which includes hosting the second edition of Coolest Projects Tunisia for the five clubs based in Greater Tunis. Equally exciting is launching Code Clubs in direct partnership with Down's syndrome associations and disability support organisations.

By creating more opportunities for young people to learn, create, and share, DotSlash AUPTIMISME is helping build a more inclusive future for creative computing in Tunisia. (HW)

Inspired to start a Code Club in your community? Find out how you can get started at **codeclub.org**.

RUHEE SHAH

Ruhee works within the global partnerships team at the Raspberry Pi Foundation as a global partnerships coordinator. In her free time, she enjoys cooking and travelling to new places.

We are constantly being asked to evaluate new AI tools, claims, risks, and opportunities

© Pauline Wee & DAIR/betterimagesofai.org/creativecommons.org/licenses/by/4.0/

THE AI OVERLOAD

Is it all a bit too much?

AI tools are everywhere, and it sometimes feels impossible to avoid. AI products, services, and news items are relentlessly pushed at us through multiple channels. Searching for recent AI news shows that at the time of writing, Claude, Apple, and Google have all launched new or significantly upgraded AI products in the last two days. The news isn't limited to new products: there's a seemingly endless stream of stories about the impact AI tools are having and will have on society, with every good news story seemingly swamped by many more negative takes.

The challenge isn't simply that AI systems exist everywhere; it's that we're being asked to constantly evaluate new claims, risks, and opportunities. I don't know about you, but even as a self-confessed tech enthusiast, sometimes this all feels a bit too much.

If this relentless march risks overwhelming a reasonably savvy educational technology enthusiast, what chance do the less invested have? Your students will come with a range of perceptions of the role of technology in their lives, so it's vital that you, as an educator, can help them navigate this constantly shifting world of AI.

Debunking the AI marketing

There are things you can do and strategies you can advocate to help your students. The first thing to consider is that the scope of AI technologies and tools is always changing. Even though the term 'AI' is now used ubiquitously, it is not new.

Not all computer tech utilises AI tools, and some that do feature little more than automation, rules-based systems, or conventional software wrapped in AI branding. Marketeers will often make AI claims to make their product sound advanced and high-tech. You should encourage students to look beyond product claims and consider whether AI-labelled tools are actually what they describe themselves to be.

Secondly, a lot of the content that is out there about AI tools is not accurate, trustworthy, transparent, or useful. Encourage your students to be discerning when consuming content, which itself may or may not be AI-generated. If you think a source is unreliable, or a claim is unlikely, don't be afraid to be curious or critical. AI tools do not always generate accurate results — far from it. In practice, this might mean encouraging students to question AI-generated answers, compare multiple sources, or discuss the claims made by AI-powered products.

Prioritising human connection

Thirdly, and possibly most importantly, remember that much of life is not, and never will be, AI-powered or AI-enabled. While AI products may play a growing role in communication and companionship, human relationships remain central to well-being, learning, and community. Nature is the result of millions of years of evolution, not a few years of AI. Go outdoors, breathe in fresh air, smell the flowers, and don't do it because an AI bot has told you to (this has been written by me, not an LLM — honest!). If you still want to get your fix of media, you can still do that without consuming AI-influenced content — read a book or watch a movie. You'll be surprised how quickly you can reconnect with old habits.

Finding balance in the AI era

If all this advice feels familiar, that's probably because it is. Similar advice has been given in the past around limiting screen time or filtering fake news. As with these aspects of tech, the way to engage with AI in a healthy and measured way is to use it in moderation, be critical, and advocate responsible use.

Beyond technology, what makes us human — our emotional connections, empathy, and communication skills should always be the most valuable and should not be passed off to machines.

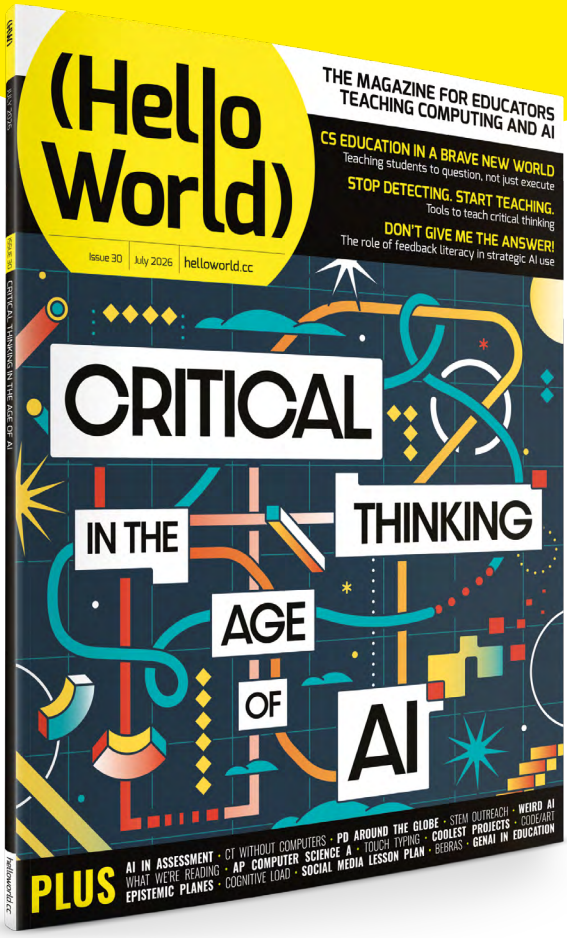
AI technology may have become increasingly woven into everyday life, but that doesn't mean it has to dominate our attention. Helping students develop critical habits, balanced perspectives, and confidence in their own judgement may ultimately be more important than keeping up with every new AI tool. (HW)

BEN HALL

Ben is a senior learning manager at the Raspberry Pi Foundation, where he leads the team developing the Certificate in Applied Computing.

SUBSCRIBE TODAY

FREE IN PRINT for UK-based educators


(Hello World)
THE MAGAZINE FOR EDUCATORS TEACHING COMPUTING AND AI
CS EDUCATION IN A BRAVE NEW WORLD
Teaching students to question, not just execute
STOP DETECTING. START TEACHING.
Tools to teach critical thinking
DON'T GIVE ME THE ANSWER!
The role of feedback literacy in strategic AI use
Issue 30 | July 2026 | helloworld.cc
CRITICAL THINKING IN THE AGE OF AI
PLUS AI IN ASSESSMENT · CT WITHOUT COMPUTERS · PD AROUND THE GLOBE · STEM OUTREACH · WEIRD AI WHAT WE'RE READING · AP COMPUTER SCIENCE A · TOUCH TYPING · COOLEST PROJECTS · CODE/ART EPISTEMIC PLANES · COGNITIVE LOAD · SOCIAL MEDIA LESSON PLAN · BEBRAS · GENAI IN EDUCATION

Why subscribe?

- Never miss an issue of Hello World
- Get notified about our latest podcast episodes
- Exclusive news, the latest research findings, and in-depth features
- Free, convenient, and full of practical ideas you can use straight away

TO SUBSCRIBE VISIT:

helloworld.cc/subscribe

Continue the conversation

Hear the voices behind the stories on our podcast. Search 'Hello World Podcast' on your favourite podcast app, or watch them on YouTube or Spotify.

helloworld.cc/pod

POD SERIES Listen or watch

(Hello World)

helloworld.cc

Raspberry Pi
Foundation

Finished reading?
Please share this magazine with others.